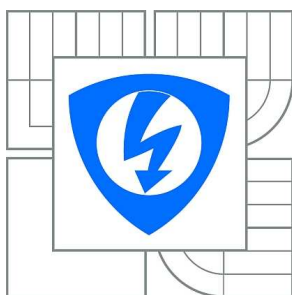


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SIMULACE ŘÍDICÍCH STRUKTUR ELEKTROMECHANICKÝCH SYSTÉMŮ

SIMULATION OF ELECTROMECHANICAL SYSTEM CONTROL STRUCTURES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

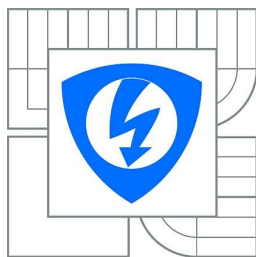
Bc. L'UBOMÍR PETRUŠKA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PAVEL VÁCLAVEK, Ph.D.

BRNO 2010



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Ľubomír Petruška

ID: 78307

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Simulace řídicích struktur elektromechanických systémů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s matematickými modely 3-fázového asynchronního motoru, 3-fázového synchronního motoru s permanentním magnetem a případně dalších typů elektrických motorů. V prostředí Matlab-Simulink vytvořte ucelenou knihovnu bloků pro modelování elektrických motorů a dalších prvků pro simulaci jejich regulačních obvodů (transformace souřadnic, SVM). Vybrané modely realizujte rovněž na procesoru řady Freescale 56F800E Hybrid Controller.

DOPORUČENÁ LITERATURA:

Caha, Z., Černý, M.: Elektrické pohony

Šubrt, J.: Elektrické regulační pohony

další dle průběžných pokynů vedoucího práce

Termín zadání: 8.2.2010

Termín odevzdání: 24.5.2010

Vedoucí práce: doc. Ing. Pavel Václavek, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Práca sa zameriava na vytváranie modelov motorov a obsahuje matematický popis činnosti asynchrónneho motora, synchronného motora s permanentnými magnetmi, jednosmerného motora s cudzím budením, jednosmerného motora so sériovým budením, jednosmerného motora s permanentnými magnetmi, spínaného reluktančného motora. Modely sú realizované na základe matematického popisu jednotlivých motorov v programe Matlab Simulink. Každý model motora sa implementuje v spojitaj aj diskkrétnej časovej variante. Vybrané modely sú implementované aj na procesore rady Freescale 56F800E Hybrid Controller. Pre každý model sa individuálne navrhne a zrealizuje grafické užívateľské rozhranie. Okrem modelov motorov je v práci popísaná metóda Space Vector Modulation s jednoduchým algoritmom pre implementovanie. Pre túto metódu je taktiež urobený model. Z vytvorených modelov je zostavená knihovna, slúžiaca pre simulovanie a testovanie riadiacich štruktúr a algoritmov. V závere práce sú prezentované výsledky modelov zo simulácií. Simulácia modelov vytvorených na procesore je taktiež realizovaná v prostredí Matlab Simulink a porovnaná s modelmi vytvorených priamo v tomto prostredí.

Kľúčové slová

Matlab Simulink, modely motorov, simulácia, asynchrónny motor, PMSM, DC motor, SRM, SVM.

Abstract

Construction of motor models is the main topic of this project. Mathematical characterization of AC machine, permanent magnet synchronous motor, separately-excited DC motor, series-wound DC motor, permanent magnet DC motor, switched reluctance motor is also described. Design of models is based on mathematical description of particular motors. Models are created in Matlab Simulink. Each model is implemented in continuous and also in discrete time variant. Selected models are implemented also on processor from Freescale 56F800E Hybrid Controller family. Each model has individual graphic user interface. Besides motor models, there is description and easy algorithm of Space Vector Modulation. Model of this method is also created. Models are build-up into a library, which can be used for simulations and tests of control structures. Results of models simulations are presented at the end of this project. Simulation of models that are implemented on processor is also made in Matlab Simulink environment and is compared to simulation of models that are implemented directly in Matlab Simulink.

Keywords

Matlab Simulink, models, simulation, AC machine, PMSM, DC motor, SRM, SVM.

Bibliografická citácia

PETRUŠKA, L.: *Simulace řídicích struktur elektromechanických systémů.*

Diplomová práce. FEKT VUT v Brně, 2010, 90s.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma *Simulace řídicích struktur elektromechanických systémů* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: 24. května 2010

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu diplomové práce doc. Ing. Pavlovi Václavkovi, PhD. za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce.

V Brne dňa: 24.mája 2010

.....
podpis autora

OBSAH

1. ÚVOD	9
2. CIELE, POŽIADAVKY A PRÍSTUP K TVORBE MODELOV	10
2.1 Vybrané modely	10
2.2 Všeobecné požiadavky na modely	11
2.3 Tvorba modelu pomocou blokov	13
2.4 Tvorba modelu cez S-function	14
2.5 Uživatelské rozhranie pomocou masiek	17
2.6 Uživatelské rozhranie cez GUI	17
2.7 Realizácia modelu na procesore	19
3. MATEMATICKÝ POPIS	21
3.1 Transformácia súradníc	21
3.2 Matematický popis asynchrónneho motora	24
3.3 Matematický popis synchrónneho motora	31
3.4 Matematický popis jednosmerného motora	35
3.5 Spínaný reluktančný motor	38
3.6 Algoritmus space vector modulation	41
4. TVORBA MODELOV	49
4.1 Návrh užívateľského rozhrania pre modely	49
4.2 Model asynchrónneho motora	54
4.3 Model synchrónneho motora s pm	56
4.4 Model DC motora	56
4.5 Model spínaného reluktančného motora	60
4.6 Model space vector modulation	63
4.7 Ďalšie modely	64
5. IMPLEMENTÁCIA VYBRANÝCH MODELOV NA PROCESORE	66
5.1 Prepojenie modelu na procesore s prostredím Matlab Simulink	66
5.2 Model DC motora s PM	67
5.3 Model synchrónneho motora s PM	70
6. SIMULÁCIA MODELOV	72
6.1 Simulácia modelov SVM a asynchrónneho motora	72

6.2 Simulácia modelu DC motora s cudzím buđením	74
6.3 Simulácia modelu DC motora so sériovým buđením	75
6.4 Simulácia modelu DC motora s permanentými magnetmi	76
6.5 Simulácia modelu PMSM	78
6.6 Simulácia modelu SRM	80
7. ZÁVER.....	83
8. ZOZNAM LITERATÚRY	84
ZOZNAM SKRATIEK A SYMBOLOV	86
ZOZNAM PRÍLOH.....	90

1. ÚVOD

V současnosti sa objavuje mnoho rôznych algoritmov pre riadenie a reguláciu pohonov. Vytvorenie modelu motora, výkonovej elektroniky, regulačných štruktúr umožňuje hodnoverne popísať správanie daného objektu. Modely na vysokej úrovni popisujú správanie a deje pri takmer ľubovoľných podmienkach, ktoré by prakticky nebolo možné realizovať na reálnych sústavách. Modely implementované na číslicových počítačoch, sú značne odolné a preto je možné na nich skúšať a testovať nové algoritmy, prípadne ich využiť k vzdelávaciemu procesu. Ďalšou ich výhodou je sledovanie dejov a veličín v inom časovom merítku. Napr. odozvu pre dej trvajúci hodiny, či roky je k dispozícii po niekoľkých sekundách.

Táto práca sa hlavne zameriava na modely motorov, ktoré budú implementované v programe Matlab Simulink. Modely budú následne doplnené o užívateľské rozhranie, aby práca s nimi bola jednoduchá a prehľadná. Cieľom je vytvoriť knihovňu takýchto modelov, ktoré by sa následne dali použiť pre simuláciu riadenia pohonov.

Vybrané modely sa pokúsime realizovať aj na procesore rady Freescale 56F800E Hybrid Controller. Predpokladáme, že takto vytvorený model bude schopný v reálnom čase dávať na výstupe rovnaké hodnoty ako skutočný motor, a že bude možné fyzicky nahradiť motor týmto modelom pre testovacie účely.

2. CIELE, POŽIADAVKY A PRÍSTUP K TVORBE MODELOV

2.1 VYBRANÉ MODELÝ

Prvým modelom motora bude asynchrónny motor. Asynchrónny motor patrí k jedným z najznámejších typom elektromotorov. Má jednoduchú konštrukciu, je robustný a spoľahlivý. Existuje preň viacero matematických popisov, podľa toho aké vlastnosti a deje potrebujeme skúmať, v rôznych súradnicových systémov. Preto sa pokúsime vytvoriť model tak, aby si do istej miery mohol užívateľ zvoliť typ modelu a užívateľské rozhranie bolo dostatočne prehľadné.

Nová technológia zliatin a výroby permanentných magnetov (PM) umožnila zmenšenie rozmerov synchronného motora a rozšírenie do rôznych aplikačných oblastí (napr. harddisky PC). Vo veľkej miere sa používajú pre polohové servomechanizmy, kde sa minimálne prejavuje ich najväčší hendikep, ktorým je nutnosť znalosti polohy rotora. Oproti asynchrónnym motorom dosahujú rovnaký výkon pri menšej váhe a rozmeroch.

Jednosmerné elektromotory patria k najstarším používaným motorom. Pre ich jednoduché riadenie vzhľadom k ostatným druhom motorov majú stále veľké uplatnenie hlavne v regulačnej technike. Preto ďalšími vytvorenými modelmi budú modely jednosmerného motora s cudzím budením, so sériovým budením a s permanentnými magnetmi.

V oblasti pohonov nachádza čoraz väčšie uplatnenie aj spínaný reluktančný motor (Switched Reluctance Motor - SRM) či už pre pohony v letectve, obrábacích strojov, pohony ventilátorov a čerpadiel. Jeho prednosťami sú: vysoký pomer momentu a zotrvačných hmôt, široký rozsah regulácie rýchlosti a dobrá riaditeľnosť.

Space Vector Modulation (SVM) je metóda pre efektívne riadenie výkonovej elektroniky pulzne šírkovou moduláciou (PWM). Na základe algoritmu [18] bude vytvorený model SVM s užívateľským rozhraním.

Pre následné simulácie a testovanie sú potrebné aj ďalšie bloky modelov ako napr. napájacie zdroje, transformácie a pod. Preto aj tieto bloky budú vytvorené

v prostředí Matlab Simulink a budú pridané do knižnice k modelom a mohli tak byť dostupné aj pre ostatných užívateľov.

2.2 VŠEOBECNÉ POŽIADAVKY NA MODELY

Dôraz pri tvorbe modelov bol kladený hlavne na funkčnosť, prehľadnosť a jednoduché používanie modelu, aby s modelom mohol pracovať aj užívateľ, ktorý nemá hlbšiu znalosť o vnútornom mechanizme.

2.2.1 Varianty modelu

Jednou z prvých požiadaviek na vytvorené modely bolo, aby sa každý model vytvoril v piatich variantoch. Každý variant má svoje výhody, či už z hľadiska prístupu k hodnotám jednotlivých veličín, časovej náročnosti výpočtu, alebo pre následnú modifikáciu. Vybrané varianty boli zvolené a pomenované nasledovne:

- **Block**

Celý model vytvoriť pomocou základných blokov Matlab Simulink [1]. Hlavnou výhodou takéhoto modelu je, že aj graficky reprezentuje model, dá sa rozdeliť do dielčích častí a tým prehľadnejšie reprezentovať jednotlivé vlastnosti a problémy. Asi najväčšou výhodou je jednoduché a ľahko prístupne pracovanie so signálmi a blokmi, či už z dôvodu tvorby, kontroly, alebo sledovania signálu – veličiny.

- **M-function continous**

Model vytvorený cez knižnicu *User-defined Functions* ako *Level-2 M-file S-function*. Využíva priamo syntaxu jazyka Matlab, a všetky jeho výhody (napr. interpolačné metódy, matematické operácie s maticami). Preto sa takýto model ľahko vytvára a odlaďuje. Model používa spojité stavové premenné.

- **M-function discrete**

Taktiež model vytvorený cez knižnicu *User-defined Functions* ako *Level-2 M-file S-function*, ktorý ale používa diskrétné stavové premenné. Oproti modelu pracujúceho so spojitými stavovými veličinami, pribudne ďalší parameter definujúci periódu vzorkovania. Model následne používa Eulerovú metódu pre aktualizovanie stavových premenných.

- **C-function continous**

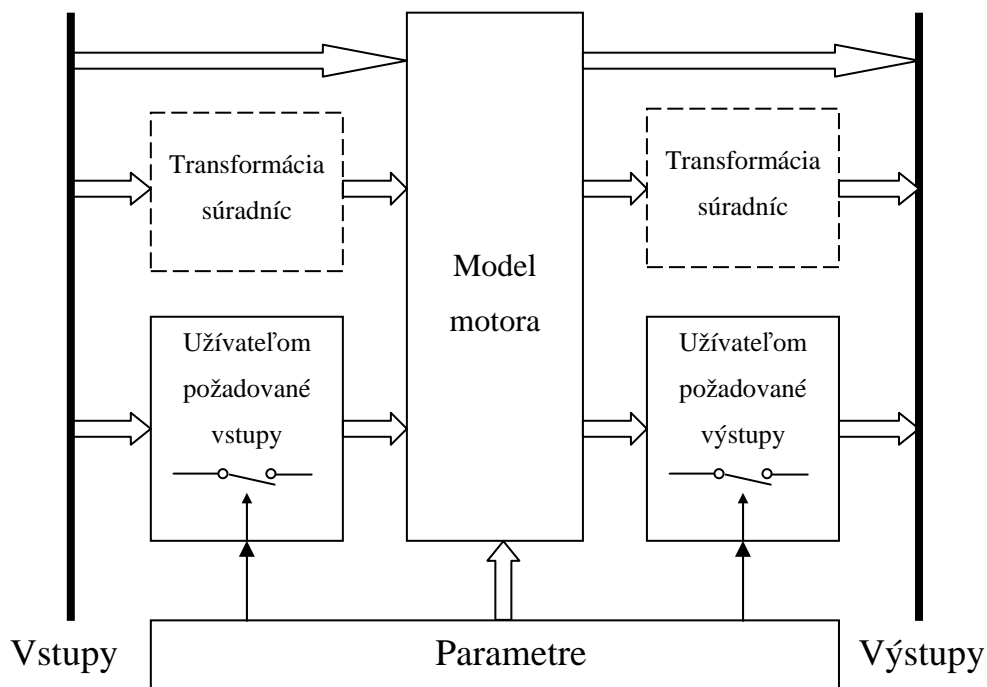
Model vytvorený cez knižnicu *User-defined Functions* ako *S-function Level-2* v jazyku C, ktorá používa spojité stavové premenné. Takto vytvorený model je značne rýchlejší než model vytvorený cez M-file.

- **C-function discrete**

Opäť model vytvorený cez knižnicu *User-defined Functions* ako *S-function Level-2* v jazyku C, ale ktorý používa diskrétné stavové premenné. Tento model následne používa Eulerovú metódu pre výpočet stavových premenných, podľa parametru periódy vzorkovania.

2.2.2 Voľba užívateľom vstupov a výstupov z modelu

Ďalšou požiadavkou bolo, aby si užívateľ modelu mohol sám vybrať z predefinovaných vstupov a výstupov, podľa jeho potreby. Model bude obsahovať určité pevné vstupy a výstupy, ktoré k nemu neodmysliteľne patria. Napr. pre vstup - napájacie napätie a moment záťaže, pre výstup otáčky rotora.



Obrázok 2.1: Bloková schéma pre zostavenia modelu motora

Východzí model je navrhnutý na predpoklade, že parametre modelu sú konštantné počas celej simulácie a preto nie je nutné, aby predstavovali samostatné

vstupy. Ak však užívateľ, potrebuje pri simulácii parameter, ktorý nie je konštantný (napr. funkcia teploty, času a pod), vytvorí sa preň samostatný vstup. Výstupy, ktoré užívateľ nepoužíva, zbytočne zneprehľadňujú model a navyše je nutné ich ošetriť.

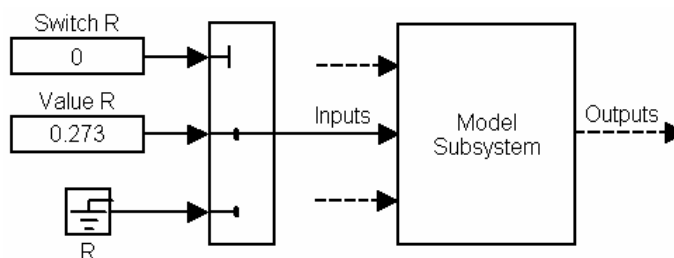
Pri tvorbe modelu motora nezávislé na variante, bola na základe požiadaviek, zostavená a použitá bloková schéma podľa obrázku 2.1. Do modelu vstupujú pevné vstupy, ktoré sa prípadne ešte transformujú (Parkova transformácia) a vstupy vybrané užívateľom namiesto parametrov. Obdobne výstupy sa skladajú z vopred definovaných pevných výstupov a výstupoch zvolených užívateľom. Vytváranie vstupov a výstupov sú stále zoradené v rovnakom poradí, nezávisle na poradí v akom boli zvolené užívateľom. To zamedzí, aby model s rovnakými vstupmi a výstupmi v zhodných, či rôznych simulačných schémach mal rôzne zoradené vstupy a výstupy, čím by mohla ľahko nastať chyba pri zapojení bloku.

2.2.3 Počiatočné hodnoty stavových premenných

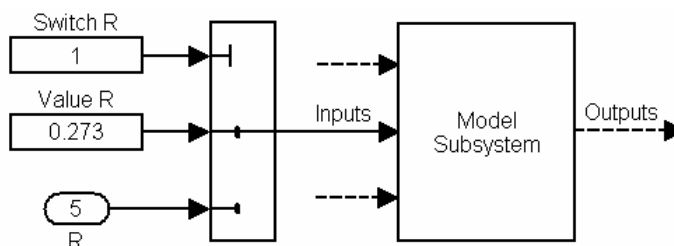
Aby užívateľ mal model plne pod kontrolou, musí mať prístup k počiatočným podmienkam stavových veličín. Preto k parametrom, ktoré nastavuje užívateľ, budú pridané aj tieto podmienky. Zároveň je z týchto podmienok vidieť, aké model používa stavové veličiny.

2.3 TVORBA MODELU POMOCOU BLOKOV

Celý model sa zostaví zo základných blokov knižnice Matlab Simulink, ktorá by mala odpovedať blokovej schéme modelu, vytvorenej podľa stavových rovníc. Bloky sa prepoja pomocou spojnic reprezentujúce signály, pričom signály môžu byť aj viacrozmerné (napr. komplexné signály). Zo zoskupenia blokov, ktoré tvoria model, sa následne vytvorí jeden samostatný blok (Subsystem). Matlab umožňuje aj programátorský prístup k jednotlivým blokom, ich parametrom a nastavení. Môžeme modifikovať parametre bloku ako aj pridávať, nahradzovať a mazať ľubovoľné bloky v zoskupení. To využijeme pre definovanie požadovaných vstupov a výstupov užívateľom.



Obrázok 2.2: Pre Subsystem modelu sa použije hodnota parametru Value R



Obrázok 2.3: Pre Subsystem modelu sa použije nový vstup R

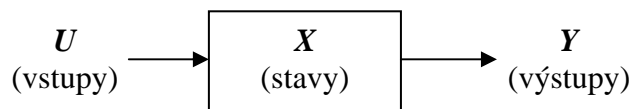
Na obrázku 2.2 a 2.3 je znázornený prístup k modelu k parametrom. Ak sa pre parameter *Resistance* požaduje hodnota zadaná užívateľom (Value R), tak sa pre prepínač signálov (Switch R) nastaví hodnota 0 (obr. 2.2). Ak sa má použiť nový vstup R, prepínač signálov sa nastaví na hodnotu 1 a nahradí sa predchádzajúci blok R (ground) blokom pre vstupný signál (obr. 2.3). Zároveň sa definuje aj jeho poradie vo vstupoch do modelu, teda aj bloku (5).

Samozrejme bolo by možné nahradzovať priamo blok vstupu R, blokom s hodnotou Value R bez použitia prepínača Switch R. To by však znamenalo stratu informácie hodnoty parametru Value R, pri nahradení vstupom R. Pri tejto koncepcii sa užívateľ môže kedykoľvek vrátiť k použitiu pôvodnej hodnoty Value R. Priame nahradenie bloku, bez použitia prepínača, je použité pre výstupy z modelu. Podľa potreby sa nahradzuje blok pre výstup (Output) ukončovacím blokom (Terminator).

2.4 TVORBA MODELU CEZ S-FUNCTION

S-function [1] predstavuje zvláštne rozhranie s podporou pre numerické riešenie diferenciálnych rovníc. Kód S-function je možné napísať programovacími jazykmi C, C++, Ada, Fortran a samozrejme Matlab. Jazyky C, C++, Ada, Fortran sú

kompilované nástrojom *mex*, pretože nie sú interpretované priamo jadrom programu Matlab Simulink.



Obrázok 2.4: Blok S-function

Blok S-function obsahuje vnútornú množinu stavov X , do ktorej vstupujú vstupy U . Pomocou S-function je možné modelovať spojité, diskkrétne aj hybridné systémy. Je výhodné ju použiť pre modelovanie dynamických systémov, čo vyplýva aj z rovníc matematicky popisujúcich S-function. Výstupy Y z S-function sú určené funkciou stavov, vstupov a simuláčného času.

$$Y = f_0(t, X, U) \quad (2.1)$$

Stavy sú určené diferenciálnou rovnicou prvého radu v prípade spojitého systému, alebo diferenčnou rovnicou v prípade diskrétného systému.

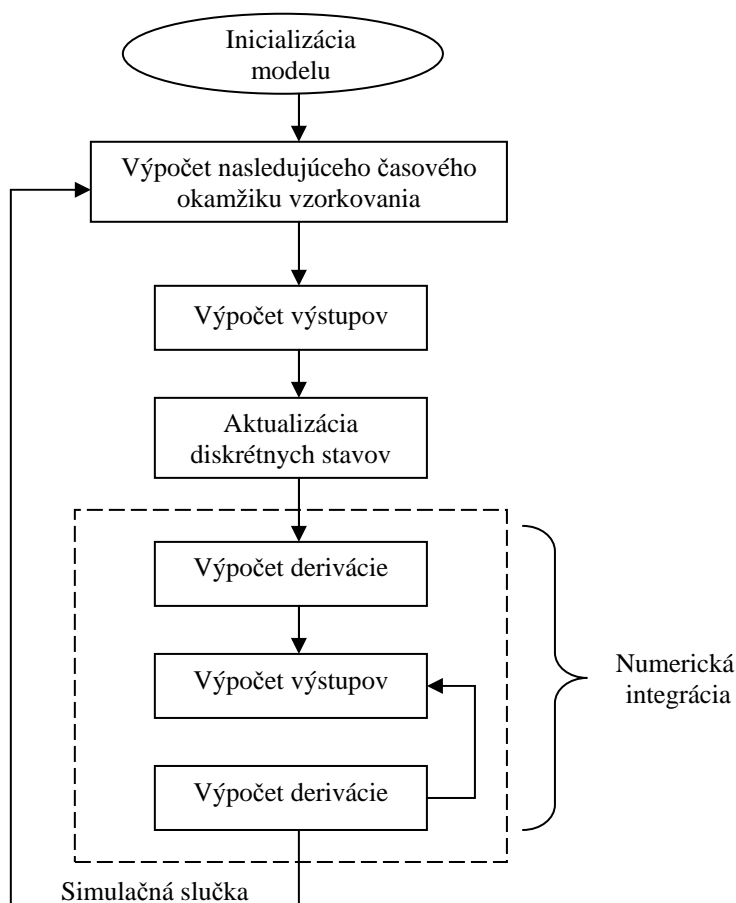
$$\dot{X}_c = f_d(t, X, U) \quad (2.2)$$

$$X_{d_{k+1}} = f_u(t, X_c, X_{d_k}, U) \quad (2.3)$$

Už z uvedených rovníc vyplýva, že kód S-funkcie sa vykonáva sekvenčne. Postupnosti jednotlivých krokov pri simulácii sú na blokovej schéme (obr. 2.5).

- **Inicializácia** – Prebehne hneď po spustení simulácie, prípadne pri zmene parametrov funkcie. Zabezpečuje správne začlenenie S-function do simulácie. Nastaví sa počet a rozmer vstupných a výstupných signálov a stavových premenných. Alokuje pamäť pre stavy a nastaví sa perióda vzorkovania. Zavolajú sa ďalšie funkcie pre správny chod a nastavenie programu. Napr. nastaví sa počiatočné hodnoty stavov.
- **Výpočet ďalšieho okamžiku vzorkovania** – Použije sa iba ak sa jedná o model s premenou dobou vzorkovania.
- **Výpočet výstupov** – Po ukončení výpočtu sa na výstupe objavia platné hodnoty pre daný časový okamžik vzorkovania.
- **Výpočet diskrétnych stavov** – Aktualizujú sa hodnoty diskrétnych stavových premenných modelu.

Numerická integrácia – Aktualizujú sa spojité stavové premenné. Použitá numerická integrácia závisí od metódy integrácie zvolenej pre simuláciu.



Obrázok 2.5: Postupnosť simulačných krokov S-function

Jeden prechod všetkými krokmi sa nazýva jeden simulačný krok. Opakujúce sa simulačné kroky tvoria simulačnú slučku. Pre každý krok podľa obr. 2.5 je určená celá rada funkcií pre korektné fungovanie programu, prístupu k parametrom, stavom, vstup, výstupom apod.

Implementovanie modelu cez S-function je jednoduchšie pomocou jazyka Matlab, ako pomocou jazyka C. Okrem ľahšej práce s dátami je možné v S-function v jazyku Matlab priamo využiť všetky funkcie a výhody programu Matlab (napr. interpolačné metódy). Tým vzniká kratší a prehľadnejší kód. Taktiež Nevznikajú chyby, ktoré sú časté pri programovaní v jazyku C (ukazovatele, zámena dátových typov integer – float). Ďalšou výhodou je, že napísanú S-function nie je nutné

kompilovat'. Oproti funkcí napísanej v jazyku C, má však značne väčšiu časovú náročnosť výpočtu.

2.5 UŽIVATELSKÉ ROZHRAŇIE POMOCOU MASIEK

Jednou z najľahších možností ako vytvoriť rozhranie pre model v prostredí Matlab Simulink je pomocou masiek (*Mask Subsystem*). Nadefinujú sa premenné a ich typ (editačné okno, zaškrtavacie pole, alebo výber 1 z N možností). Ďalej je možné ku každému parametru užívateľského rozhrania napísať *Callback* funkciu, kde sa naplnia, prepočítajú, či ošetrí parametre. V maske sa taktiež definuje vzhľad bloku, ktorý reprezentuje model. Názvy vstupov a výstup s použitím znakov gréckej abecedy. Grafický dizajn bloku (textové pole, pozadie, obrázok, grafy ...)

Vytvorenie užívateľského rozhrania týmto spôsobom je síce nenáročné, avšak má aj značné nevýhody, ktoré sa prejavujú hlavne pri väčších modeloch s mnohými parametrami. Zásadným problémom je, nemožnosť nadefinovať funkciu pre tlačidlá Apply, alebo OK. To spôsobuje, problémy pri predávaní parametrov modelu, kedy je nutné mať dáta a parametre plne pod kontrolou po celý čas. Ďalšou nevýhodou je neprehľadnosť pri modeli s veľkým počtom parametrov. Maska neumožňuje zvoliť rozmiestnenie a veľkosť jednotlivých komponent v užívateľskom rozhraní. Komponenty sú zoradené v poradí, ktoré je možné nadefinovať, ale skladá ich pod seba, čím vzniká dlhý zoznam položiek, v ktorom treba listovať. V niektorých prípadoch je vhodné doplniť užívateľské rozhranie o ďalšie komponenty ako tlačidlá, tabuľky, grafy, ale to cez masku sa nedá vytvoriť. Preto sa tento spôsob užívateľského rozhrania použije iba pre jednoduché modely, funkcie, alebo transformácie.

2.6 UŽIVATELSKÉ ROZHRAŇIE CEZ GUI

2.6.1 Základný princíp

Všetky nedostatky, ktorá malo vytvorenie užívateľského rozhrania cez masku subsystemu, sa dajú odstrániť použitím GUI (Graphical User Interface) *Layout Editor*. Cez editor je možné vkladať všetky bežne používané komponenty, vložiť menu, vytvoriť viac dialógových okien vzájomne prepojených. Ľubovoľne

rozmisťňovať komponenty a pritom ich presne prispôsobovať podľa potrieb. Pre každú komponentu je možné definovať spustiteľný kód v jazyku Matlab, ktorý sa vykoná ako odozva na nieaký zásah užívateľa napr. kliknutie tlačidlom myši na daný objekt, pozíciou kurzora na danom objekte, alebo pri vytváraní či ukončovaní objektu a pod. V editore je možné nastaviť ešte mnohé ďalšie vlastnosti GUI. Príkladom môže byť nastavenie, aby mohla byť spustená iba jedna inštancia GUI, nemenné rozmery okna, zarovnanie, či definovanie poradia tabulátorov komponent.

Vytvorené rozhranie pozostáva z dvoch hlavných častí. Prvou časťou je samotné grafické rozhranie (Fig-file). Druhou neoddeliteľnou časťou je zdrojový kód (M-file), kde sú uložené odozvy pre jednotlivé komponenty a prepojenie s konfiguráciou na jemu odpovedajúce grafické rozhranie. Tento kód je v prípade používania GUI editoru generovaný automatický programom Matlab a stačí dopísať funkcie pre jednotlivé odozvy.

Vlastnosti pre každý pridaný komponent a aj pre samotné hlavné okno GUI, sa dajú nastaviť v okne *Property Inspector*. Všetky vlastnosti, ktoré sa týkajú danej komponenty sú prístupné práve cez toto okno, čím je celá práca vytvárania grafického rozhrania rýchla a efektívna. Dôležitým parametrom je položkou *Tag*, ktorá je názvom pre komponentu a súčasne sa cez ňu definujú funkcie pre spätné volania odoziev.

2.6.2 Práca s dátami a funkciami GUI

Matlab používa abstraktnú triedu, ktorá je nad všetkými ostatnými triedami – *handle*. *Handle* predstavuje identifikátor objektu a pre každý objekt je jedinečný. To ma za následok, že sme schopní presne určiť s koľkými objektmi pracujeme a vieme sa odkázať na hociktorý aktívny objekt.

V M-file GUI sa taktiež pracuje s týmito identifikátormi. Pre celý objekt GUI sa vytvorí štruktúra *handles*, v ktorej sú identifikátory všetkých použitých komponentov grafického rozhrania. Táto štruktúra existuje iba lokálne, počas existencie rozhrania. V každom kroku vo vnútri funkcie dokážeme pracovať s ľubovoľným objektom, zistiť a nastaviť jeho atribúty.

Každá spätná funkcia od komponenty – *Callback* je volaná s tromi parametrami:

- *hObject* – je identifikátor objektu, ktorý funkciu vyvolal. Pre prácu s dátami a vlastnosťami danej komponenty, pristupujeme priamo cez tento identifikátor.
- *eventdata* – reprezentuje sekvenciu udalostí vyvolaných užívateľom. Niektoré komponenty predávajú aj udalosti, ktoré vyvolali alebo predchádzali spusteniu tejto funkcie. Napr. pohyb kurzora, aká klávesa bola stlačená apod. Funkciám, ktoré nepracujú s nimi nepracujú sa predá prázdny argument.
- *handles* – zoznam všetkých identifikátorov komponentov GUI, ako aj identifikátor samotného aktívneho užívateľského rozhrania, a prípadne aj prístup k dátam.

Pri vyvolaní spätnej funkcií od komponenty existujú iba tie dáta, ktoré sú predané do funkcie cez *handle*, alebo ak sú obsiahnuté priamo v komponente a samozrejme dáta v *baseworkspace*. Pridaním dát do *baseworkspace*, ak to nie je úmyselné, by sme zbytočne pridávali dočasné premenné, ktoré by sme museli následne pri ukončení užívateľského rozhrania odstrániť. To by nebolo efektívne obzvlášť, keď Matlab ponúka lepšie riešenia.

Jedným z riešení je využiť priestor *UserData*, ktorá je súčasťou každej komponenty. Prístup k dátam je teda podobný, ako k hocakému inému parametru danej komponenty. Tento prístup je skôr vhodný, ak sa dáta týkajú priamo tejto komponenty. V opačnom prípade, by tento prístup vniesol do programu chaos.

Ďalšou možnosťou práce s dátami, ktorá bola aj aplikovaná v modeloch, je pridať dáta do štruktúry *handles*. Takto je možné zdieľať, pridávať a modifikovať dáta v hocakej funkcii. Pridané dáta do štruktúry *handles* sú v rámci funkcie ako lokálne premenné. Dôležitou operáciou je vždy modifikované dáta uložiť, pred ukončením funkcie, lebo inak by zanikli. Podobne, ak je niekde návrat z funkcie, musia sa dáta (*handles*) znovu načítať, aby sa prejavila prípadná modifikácia.

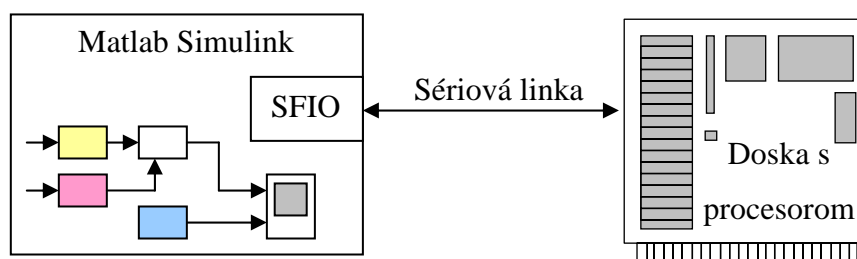
2.7 REALIZÁCIA MODELU NA PROCESORE

Vybrané modely sa pokúsime na implementovať na procesore rady Freescale 56F800E Hybrid Controller. Ide o procesor, ktorý pracuje v tzv. *zlomkovej aritmetike* (fractional arithmetic) [2], [3], v ktorej sa číslo x môže pohybovať iba v intervale $\langle -1; 1 - 2^{-M} \rangle$, kde M je dĺžka slova (počet bitov b).

$$x = (-1)b_0 + \sum_{m=1}^M b_m 2^{-m} \quad (2.4)$$

Záporné čísla sú reprezentované v druhom doplnku (b_0 - MSB, b_M - LSB). Výsledne hodnoty z operácii, ktoré sú mimo daný interval, bývajú najčastejšie *obmedzené* (saturated), alebo *pretečú* (overflow). Z toho dôvodu je nutné všetky vstupy do procesoru normalizovať – upraviť ich hodnoty tak, aby neprekročili rozsah $\langle -1;1 \rangle$ a obdobne denormalizovať výstupy z procesoru, aby správne číselne vyjadrovali danú veličinu. Taktiež vo vnútri modelu, dbať na tento fakt a korektne pracovať s jednotlivými premennými.

Kód modelu pre procesor bude programovaný pomocou programu CodeWarrior IDE. Pre prepojenie modelu a simuláciu využijeme toolbox SFIO programu Matlab Simulink vyvinutý na UAMT FEKT v Brne. Tým získame možnosť pracovať a simulovať tieto modely spolu s ostatnými modelmi, ktoré sú realizované priamo v prostredí Matlab Simulink, používať rovnaké vstupne signály a mnoho ďalších výhod vyplývajúcich z tohto prepojenia.



Obrázok 2.6: Prepojenie procesoru s programom Matlab Simulink cez SFIO

3. MATEMATICKÝ POPIS

3.1 TRANSFORMÁCIA SÚRADNÍC

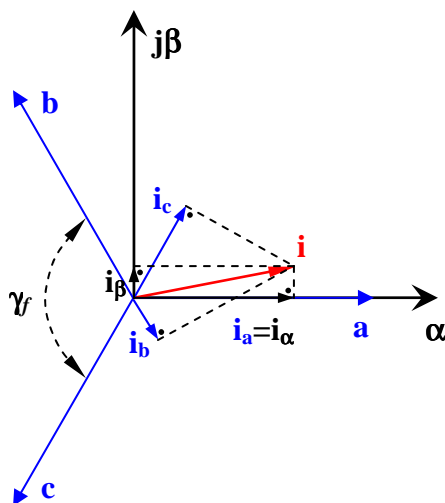
Pri odvodení modelu a v ďalších častiach práce (asynchrónny, synchronný motor) budú využité transformácie súradníc. Preto si uvedieme základné transformácie, ktoré sa dajú efektívne aplikovať pre prechod medzi popismi súradníc

3.1.1 Clarkovej transformácia

Umožňuje preniesť tri osi dvojdimenzionálneho súradnicového systému statora do dvoch osí ($\alpha\beta$) pri zachovaní súradnicového systému [4]. Keďže väčšina trojfázových motorov má zapojenie vinutí do hviezdy, splňujú podmienku, z ktorej je odvodená Clarkovej transformácia:

$$i_a(t) + i_b(t) + i_c(t) = 0 \quad (3.1)$$

kde i_a , i_b , i_c sú satorové prúdy.



Obrázok 3.1: Clarkovej transformácia

Transformácia z trojfázového do dvojfázového systému ($\alpha\beta$) je zapísaná v nasledujúcej matici, kde γ_f je uhol medzi fázami, teda $\gamma_f = 2\pi/3$

$$\begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} = K_{tr} \begin{bmatrix} 1 & \cos(\gamma) & \cos(2\gamma) \\ 0 & \sin(\gamma) & \sin(2\gamma) \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} \quad (3.2)$$

$$\begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cos(\gamma) & \sin(\gamma) \\ \cos(2\gamma) & \sin(2\gamma) \end{bmatrix} \begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix}. \quad (3.3)$$

Konštanta K_{tr} sa volí spravidla zvyčajne 2/3 vzhľadom k zjednodušenému výpočtu pri práci s komplexorom. Transformácia podľa vzťahu (3.2) a (3.3) sa nazýva Forward Clarke Transform a Reverse Clarke Transform. Obdobné vzťahy platia aj pre napätia a magnetické toky

Pre praktické použitie sa často používa ich upravená podoba. Keďže platí rovnica (3.1), je možné vyjadriť veličinu jednej fáze ako lineárnu kombináciu ďalších dvoch, čo v praxi znamená úsporu jedného snímača. Dosadením konštánt γ a K_{tr} , do (3.2) a (3.3) je možné transformáciu napísať v tvare

$$\begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} = \begin{bmatrix} i_a(t) \\ \frac{1}{\sqrt{3}}(i_b(t) - i_c(t)) \end{bmatrix} \quad (3.4)$$

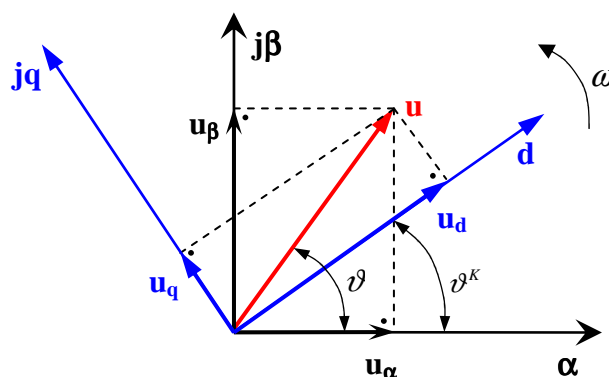
a obdobne inverzná Clarkovej transformácia

$$\begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} = \begin{bmatrix} i_\alpha(t) \\ -\frac{1}{2}i_\alpha(t) + \frac{\sqrt{3}}{2}i_\beta(t) \\ -\frac{1}{2}i_\alpha(t) - \frac{\sqrt{3}}{2}i_\beta(t) \end{bmatrix}. \quad (3.5)$$

3.1.2 Parkova transformácia

Pri natáčaní rotora sa mení magnetická väzba medzi statorovým a rotorovým vinutím AC stroja. Preto si zavedieme spoločný súradnicový systém K, aby sme mohli pozorovať deje v jednotlivých vinutiach [5]. Tým súčasne vytvoríme pevnú magnetickú väzbu s vinutím rotora. V ustálenom stave sú napätia a prúdy fiktívnych vinutí konštantnými (jednosmernými) veličinami, čím sa riešenie zjednoduší, pretože sa dajú vyjadriť prechodne striedavé veličiny pomocou týchto jednosmerných veličín.

Nech v súradnicovom systéme S ($\alpha\beta$) je uhol medzi statorovým a rotorovým vinutím je ϑ v smere otáčania rotora ω . Potom obrázok 3.2 znázorňuje transformáciu súradníc do spoločného súradnicového K (dq).



Obrázok 3.2: Transformácia do spoločného súradnicového systému K

Keďže transformácia súradníc S do spoločného súradnicového systému závisí na uhle natočenia ϑ^K (čo je zrejmé z obrázku 3.2), môžeme komplexor napätia vyjadriť pomocou súradníc S ako

$$\mathbf{u}^K = u_d + ju_q = \mathbf{u}^S e^{-j\vartheta^K}. \quad (3.6)$$

Komplexor napätia \mathbf{u}^S je tvorený zložkami u_α a u_β dosadením týchto zložky do rovnice (3.6), získame vzťah (3.7), ktorý sa nazýva Parkova transformácia.

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} \cos(\vartheta^K) & \sin(\vartheta^K) \\ -\sin(\vartheta^K) & \cos(\vartheta^K) \end{bmatrix} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} \quad (3.7)$$

Taktiež je možné urobiť transformáciu zo spoločného súradnicového systému rotora do súradnicového systému S, ktorá sa nazýva inverzná Parkova transformácia

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \begin{bmatrix} \cos(\vartheta^K) & -\sin(\vartheta^K) \\ \sin(\vartheta^K) & \cos(\vartheta^K) \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix}. \quad (3.8)$$

3.1.3 Spojenie Clarkovej a Parkovej Transformácií

Spojením Clarkovej a Parkovej transformácie je možné transformovať priamo trojfázové napätie (aj prúd, či magnetický tok) do spoločnej súradnej sústavy K v jednom kroku. Dosadením vzťahu (3.2) do transformácií (3.7) dostávame vzťah

$$\begin{bmatrix} u_d^K \\ u_q^K \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\vartheta^K) & \cos(\vartheta^K - 2\pi/3) & \cos(\vartheta^K + 2\pi/3) \\ \sin(\vartheta^K) & \sin(\vartheta^K - 2\pi/3) & \sin(\vartheta^K + 2\pi/3) \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (3.9)$$

Obdobne dosadením rovnice (3.3) do (3.8) dostaneme spätnú transformáciu zo spoločnej súradnej sústavy do jednotlivých zložiek trojfázových napätia.

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} \cos(\vartheta^K) & \sin(\vartheta^K) \\ \cos(\vartheta^K - 2\pi/3) & \sin(\vartheta^K - 2\pi/3) \\ \cos(\vartheta^K + 2\pi/3) & \sin(\vartheta^K + 2\pi/3) \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix} \quad (3.10)$$

3.2 MATEMATICKÝ POPIS ASYNCHRONNÉHO MOTORA

3.2.1 Odvodnenie základných vzťahov

Pri odvodení matematického modelu asynchrónneho motora (ASM) [6] budeme vychádzať z niektorých zjednodušujúcich predpokladov. Budeme predpokladať trojfázové súmerné harmonické napájacie napätie U_s s frekvenciou f

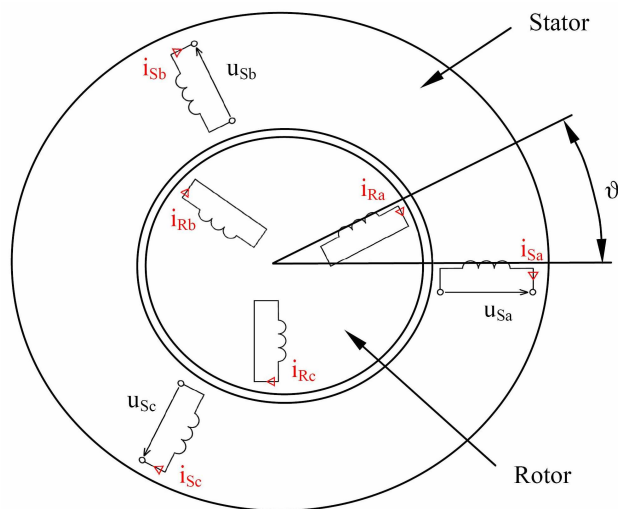
$$\begin{aligned} u_a &= U_s \cos(2\pi f t) \\ u_b &= U_s \cos\left(2\pi f t - \frac{2}{3}\pi\right) \\ u_c &= U_s \cos\left(2\pi f t - \frac{4}{3}\pi\right) \end{aligned} \quad (3.11)$$

Odpory a indukčnosti v jednotlivých fázach statorového a rotorového vinutia sú rovnaké, vinutia rotora a statora majú rovnaký počet závitov a sú priestorovo symetricky rozložené v drážkach. Ďalej budeme predpokladať lineárnu magnetizačnú charakteristiku, harmonické rozloženie magnetickej indukcie vo vzduchovej medzere a zanedbáme tepelné straty a straty v železe.

Nech rotorové vinutia sú nakrátko ako to reprezentuje obrázok 3.3 a posun rotora voči statoru je uhol ϑ . Potom využitím Clarkovej transformácie (3.2) dostávame pre napätia a prúdy vzťahy

$$\begin{aligned} u_S^S &= u_{S\alpha} + j u_{S\beta} & i_S^S &= i_{S\alpha} + j i_{S\beta} \\ u_R^R &= 0 & i_R^R &= i_{R\alpha} + j i_{R\beta} \end{aligned}, \quad (3.12)$$

kde horný index vyjadruje súradný systém statora S a rotora R.



Obrázok 3.3: Usporiadanie vinutí pri odvodzovaní matematického modelu ASM

S využitím Kirchhoffových zákonov, môžeme napísať komplexorové rovnice asynchrónneho motora

$$\mathbf{u}_S^S = R_S \mathbf{i}_S^S + \frac{d}{dt} \boldsymbol{\psi}_S^S, \quad (3.13)$$

$$\mathbf{u}_R^R = 0 = R_R \mathbf{i}_R^R + \frac{d}{dt} \boldsymbol{\psi}_R^R. \quad (3.14)$$

R_S a R_R sú odpory statora a rotora. Sprážené magnetické toky statora $\boldsymbol{\psi}_S^S$ a rotora $\boldsymbol{\psi}_R^R$ závisia na vlastnej indukčnosti vinutia statora L_S , rotora L_R a zároveň na vzájomnej indukčnosti L_m , ktorá sa mení v závislosti na uhlovej polohe rotora ϑ

$$\boldsymbol{\psi}_S^S = L_S \mathbf{i}_S^S + L_m e^{j\vartheta} \mathbf{i}_R^R, \quad (3.15)$$

$$\boldsymbol{\psi}_R^R = L_R \mathbf{i}_R^R + L_m e^{-j\vartheta} \mathbf{i}_S^S. \quad (3.16)$$

S využitím Parkovej transformácie môžeme rovnice vyjadriť v spoločnom súradnicovom systéme K. Ak uhol medzi osou statorového vinutia a spoločným súradnicovým vinutím označíme ϑ^K (pozri obrázok 3.2), tak súradnicový systém R je pootočený o uhol $-(\vartheta - \vartheta^K)$. Potom napätia a prúdy v súradnom systéme K sú

$$\begin{aligned} \mathbf{u}_S^K &= \mathbf{u}_S^S e^{-j\vartheta^K} \\ \mathbf{i}_S^K &= \mathbf{i}_S^S e^{-j\vartheta^K} \\ \mathbf{i}_R^K &= \mathbf{i}_R^R e^{-j(\vartheta^K - \vartheta)} \end{aligned} \quad (3.17)$$

Dosadením rovnic (3.13) až (3.16) do rovnic (3.17) dostávame rovnice v spoločnom súradnicovom systéme K, z ktorých je vidieť, že sprážený magnetický tok nezávisí na vzájomnej polohe vinutia statora a rotora

$$\mathbf{u}_S^K = R_S \mathbf{i}_S^K + \frac{d}{dt} \boldsymbol{\psi}_S^K + j\omega^K \boldsymbol{\psi}_S^K, \quad (3.18)$$

$$0 = R_R \mathbf{i}_R^K + \frac{d}{dt} \boldsymbol{\psi}_R^K + j(\omega^K - \omega_e) \boldsymbol{\psi}_R^K, \quad (3.19)$$

$$\boldsymbol{\psi}_S^K = L_S \mathbf{i}_S^K + L_m \mathbf{i}_R^K, \quad (3.20)$$

$$\boldsymbol{\psi}_R^K = L_S \mathbf{i}_R^K + L_m \mathbf{i}_S^K. \quad (3.21)$$

Ak zvolíme súradnicový systém K spojený so statorovým systémom S tak, že uhol ϑ^K je konštantný, čo je vhodné pre sledovanie statorových veličín, potom

$$\frac{d}{dt} \vartheta^K = \omega^K = 0 \quad (3.22)$$

a môžeme rovnice (3.8) a (3.9) upraviť na tvar

$$\frac{d}{dt} \boldsymbol{\psi}_S^K = \mathbf{u}_S^K - R_S \mathbf{i}_S^K \quad (3.23)$$

$$\frac{d}{dt} \boldsymbol{\psi}_R^K = -R_R \mathbf{i}_R^K + j\omega_e \boldsymbol{\psi}_R^K \quad (3.24)$$

Pretože moment je nezávislý na súradnom systéme je teda viac vyjadrení pre moment motora. Pre následnú simuláciu motora budeme používať vzťah pre moment motora

$$T_e = \frac{3}{2} P_p \Im \{ \boldsymbol{\psi}^{K*} \mathbf{i}^K \} = \frac{3}{2} P_p \Im \{ \boldsymbol{\psi}_S^* \mathbf{i}_S \}. \quad (3.25)$$

Medzi rýchlosťou rotora ω_m a elektrickou uhlovou rýchlosťou ω_e platí vzťah

$$\omega_e = P_p \omega_m, \quad (3.26)$$

P_p je počet pólových dvojíc motora. Pre sklz s , ktorý charakterizuje rozdiel medzi uhlovou rýchlosťou rotora a synchronnou uhlovou rýchlosťou magnetického poľa ω_s

$$s = 1 - \frac{\omega_m}{\omega_s}, \quad (3.27)$$

$$\omega_s = \frac{2\pi f}{P_p}. \quad (3.28)$$

Mechanickú uhlovú rýchlosť na základe pôsobiacich momentov pri zanedbaní trenia (tlmenia) je

$$\frac{d}{dt} \omega_m = \frac{1}{J} (T_e - T_L), \quad (3.29)$$

T_L je moment záťaže a J je moment zotrvačnosti motora. Dosadením výrazu (3.25) do rovnice (3.29) a prepočtom mechanickej uhlovej rýchlosti na elektrickú dostaneme vzťah

$$\frac{d}{dt} \omega_e = \frac{P_p}{J} \left(\frac{3}{2} P_p \Im \{ \Psi_s^* \mathbf{i}_s \} - T_L \right). \quad (3.30)$$

3.2.2 T-model

Rovnice (3.20), (3.21), (3.23), (3.24) a (3.30) predstavujú model asynchrónneho motora označovaný ako T-model. Pre prehľadnosť si ich zhrňme do nasledujúceho výrazu

$$\begin{aligned} \frac{d}{dt} \Psi_s &= \mathbf{u}_s - R_s \mathbf{i}_s \\ \frac{d}{dt} \Psi_r &= -R_r \mathbf{i}_r + j\omega_e \Psi_r \\ \Psi_s &= L_s \mathbf{i}_s + L_m \mathbf{i}_r \\ \Psi_r &= L_r \mathbf{i}_r + L_m \mathbf{i}_s \\ \frac{d}{dt} \omega_e &= \frac{P_p}{J} \left(\frac{3}{2} P_p \Im \{ \Psi_s^* \mathbf{i}_s \} - T_L \right) \end{aligned} \quad (3.31)$$

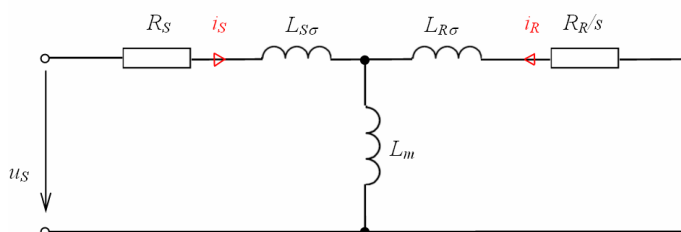
Vzťahy pre prúdy statora a rotora získame vyjadrením z rovníc (3.20) a (3.21)

$$\begin{aligned} \mathbf{i}_s &= \frac{L_r}{L_s L_r - L_m^2} \Psi_s - \frac{L_m}{L_s L_r - L_m^2} \Psi_r \\ \mathbf{i}_r &= \frac{L_s}{L_s L_r - L_m^2} \Psi_r - \frac{L_m}{L_s L_r - L_m^2} \Psi_s \end{aligned} \quad (3.32)$$

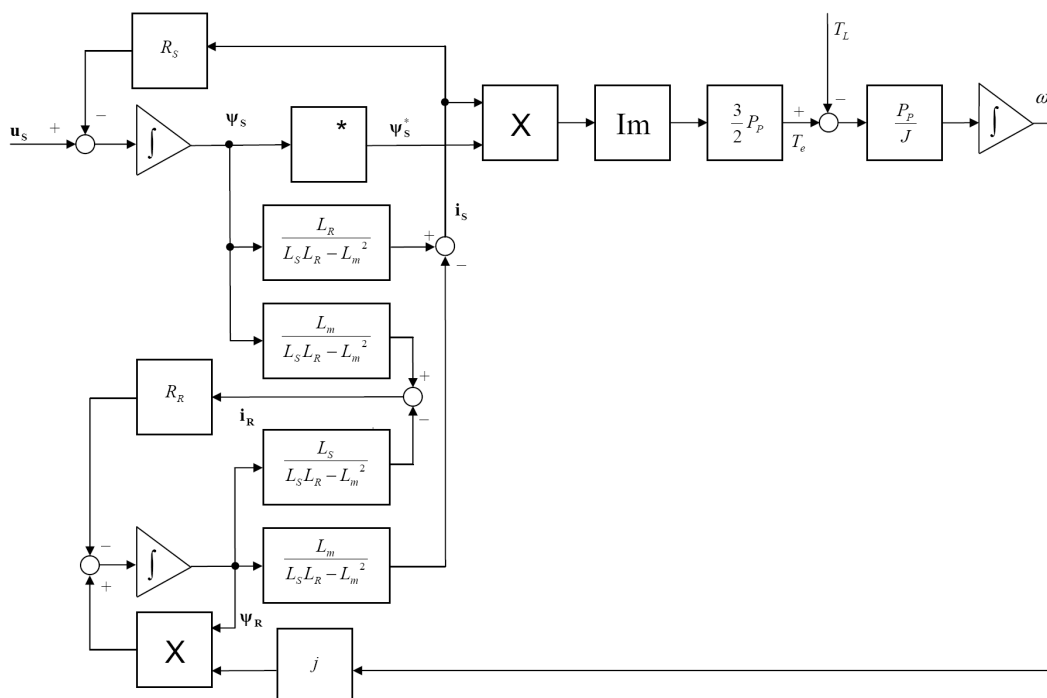
Indukčnosť statora L_s a rotora L_r je možné vyjadriť pomocou vzájomnej (magnetizačnej) indukčnosti L_m a rozptylových indukčností $L_{s\sigma}$, $L_{r\sigma}$. [7]

$$\begin{aligned} L_s &= L_{s\sigma} + L_m \\ L_r &= L_{r\sigma} + L_m \end{aligned} \quad (3.33)$$

Rovniciam (3.31) a (3.33) odpovedá náhradná elektrická schéma na obr. 3.4.



Obrázok 3.4: Náhradná elektrická schéma T-modelu asynchrónneho motora pre ustálený stav



Obrázok 3.5: Náhradná elektrická schéma T-modelu asynchrónneho motora pre ustálený stav

3.2.3 Γ -model

Tento model je výhodný hlavne preto, že neobsahuje žiaden nadbytočný parameter, ako je to u T-modelu, ktorý môže spôsobovať problémy z hľadiska identifikácie, či realizácii stavových rekonštruktorov pre riadenie. Aby bol počet parametrov minimálny zavedie sa substitúcia

$$\gamma = \frac{L_S}{L_m}, \quad (3.34)$$

$$\Psi_R = \frac{1}{\gamma} \Psi_{R\Gamma}, \quad (3.35)$$

$$\mathbf{i}_R = \mathcal{I}_{R\Gamma} \cdot \mathbf{i}_{R\Gamma}. \quad (3.36)$$

Dosadením vzťahov (3.34) až (3.36) do vzťahov (3.20) a (3.21) a po úpravách (keďže všetky rovnice sú v spoločnom súradnicovom systéme, rovnice sú bez horného indexu K) dostaneme

$$\Psi_S = L_M (\mathbf{i}_S + \mathbf{i}_{R\Gamma}), \quad (3.37)$$

$$\Psi_{R\Gamma} = \Psi_S + L_L \mathbf{i}_{R\Gamma}, \quad (3.38)$$

kde L_M je nová magnetizačná indukčnosť a L_L rozptylová indukčnosť

$$L_M = L_S = \mathcal{I}_{L_m}, \quad (3.39)$$

$$L_L = \mathcal{I}_{L_{S\sigma}} + \gamma^2 L_{R\sigma}. \quad (3.40)$$

Po dosadení (3.35) a (3.36) do rovnice (3.24) sa zmení pôvodná rovnica nasledovne

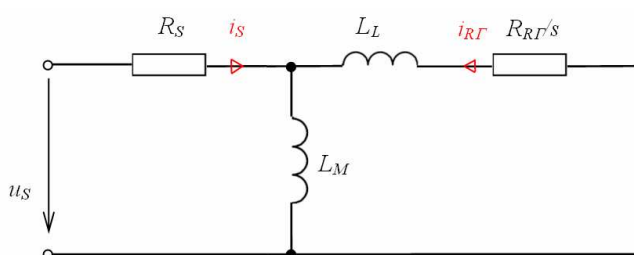
$$\frac{d}{dt} \Psi_{R\Gamma} = -\gamma^2 R_R \mathbf{i}_{R\Gamma} + j\omega_e \Psi_{R\Gamma}. \quad (3.41)$$

Vzťahy popisujúce Γ -model dostaneme na základe odvodených rovníc doplnené rovnicami T-modelu, ktoré sa nezmenili. Výraz $\gamma^2 R_R$ nahradíme novou premenou $R_{R\Gamma}$ a rovnice popisujúce Γ -model budú vyzerat'

$$\begin{aligned} \frac{d}{dt} \Psi_S &= \mathbf{u}_S - R_S \mathbf{i}_S \\ \frac{d}{dt} \Psi_{R\Gamma} &= -R_{R\Gamma} \mathbf{i}_{R\Gamma} + j\omega_e \Psi_{R\Gamma} \\ \Psi_S &= L_M (\mathbf{i}_S + \mathbf{i}_{R\Gamma}) \\ \Psi_{R\Gamma} &= \Psi_S + L_L \mathbf{i}_{R\Gamma} \\ \frac{d}{dt} \omega_e &= \frac{P_P}{J} \left(\frac{3}{2} P_P \Im \{ \Psi_S^* \mathbf{i}_S \} - T_L \right) \end{aligned} \quad (3.42)$$

Substitúcie použité pre prechod od T-modelu ku Γ -modelu

$$\begin{aligned}\gamma &= \frac{L_S}{L_m} \\ \Psi_{R\Gamma} &= \gamma \Psi_R \\ \mathbf{i}_{R\Gamma} &= \frac{1}{\gamma} \mathbf{i}_R \\ L_M &= L_S = \gamma L_m \\ L_L &= \gamma L_{S\sigma} + \gamma^2 L_{R\sigma} \\ R_{R\Gamma} &= \gamma^2 R_R\end{aligned} \quad (3.43)$$



Obrázok 3.6: Náhradná elektrická schéma Γ -modelu asynchrónneho motora pre ustálený stav

3.2.4 Inverzný Γ -model

Ak by sme zvolili iný tvar substitúcie danú výrazmi (3.44), (3.45) a (3.46), tak dospejeme k modelu, ktorý je nazvaný inverzný Γ -model, skrátene Γ^{-1} -model

$$\gamma' = \frac{L_m}{L_R}, \quad (3.44)$$

$$\Psi_R = \frac{1}{\gamma'} \Psi'_{R\Gamma}, \quad (3.45)$$

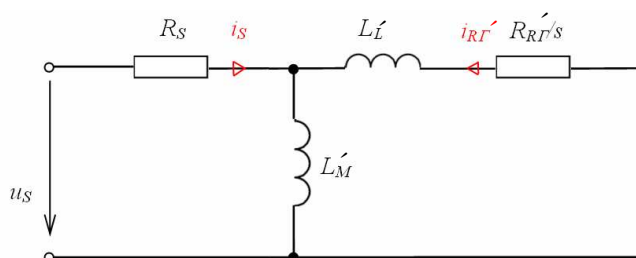
$$\mathbf{i}_R = \gamma' \mathbf{i}'_{R\Gamma}. \quad (3.46)$$

Podobným spôsobom a postupom, akým bol odvodený Γ -model na základe T-modelu dostaneme rovnice Γ^{-1} -modelu

$$\begin{aligned}
 \frac{d}{dt} \Psi_S &= \mathbf{u}_S - R_S \mathbf{i}_S \\
 \frac{d}{dt} \Psi'_{R\Gamma} &= -R'_{R\Gamma} \mathbf{i}'_{R\Gamma} + j\omega_e \Psi'_{R\Gamma} \\
 \Psi_S &= \Psi'_{R\Gamma} + L'_L \mathbf{i}_S \\
 \Psi'_{R\Gamma} &= L'_M (\mathbf{i}_S + \mathbf{i}'_{R\Gamma}) \\
 \frac{d}{dt} \omega_e &= \frac{P_p}{J} \left(\frac{3}{2} P_p \Im \{ \Psi_S^* i_S \} - T_L \right)
 \end{aligned}
 \tag{3.47}$$

Substitúcie použité pre prechod od T-modelu k Γ^{-1} -modelu

$$\begin{aligned}
 \gamma' &= \frac{L_m}{L_R} \\
 \Psi'_{R\Gamma} &= \gamma' \Psi_R \\
 \mathbf{i}'_{R\Gamma} &= \frac{1}{\gamma'} \mathbf{i}_R \\
 L'_M &= \gamma' L_m \\
 L'_L &= L_{S\sigma} + \gamma' L_{R\sigma} \\
 R'_{R\Gamma} &= \gamma'^2 R_R
 \end{aligned}
 \tag{3.48}$$

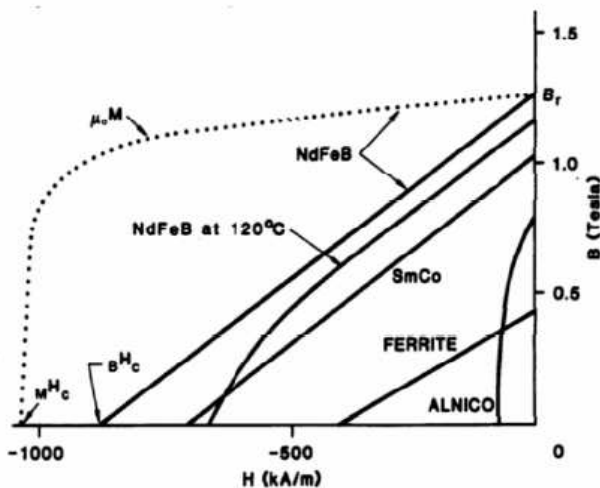


Obrázok 3.7: Náhradná elektrická schéma inverzný Γ -modelu asynchrónneho motora pre ustálený stav

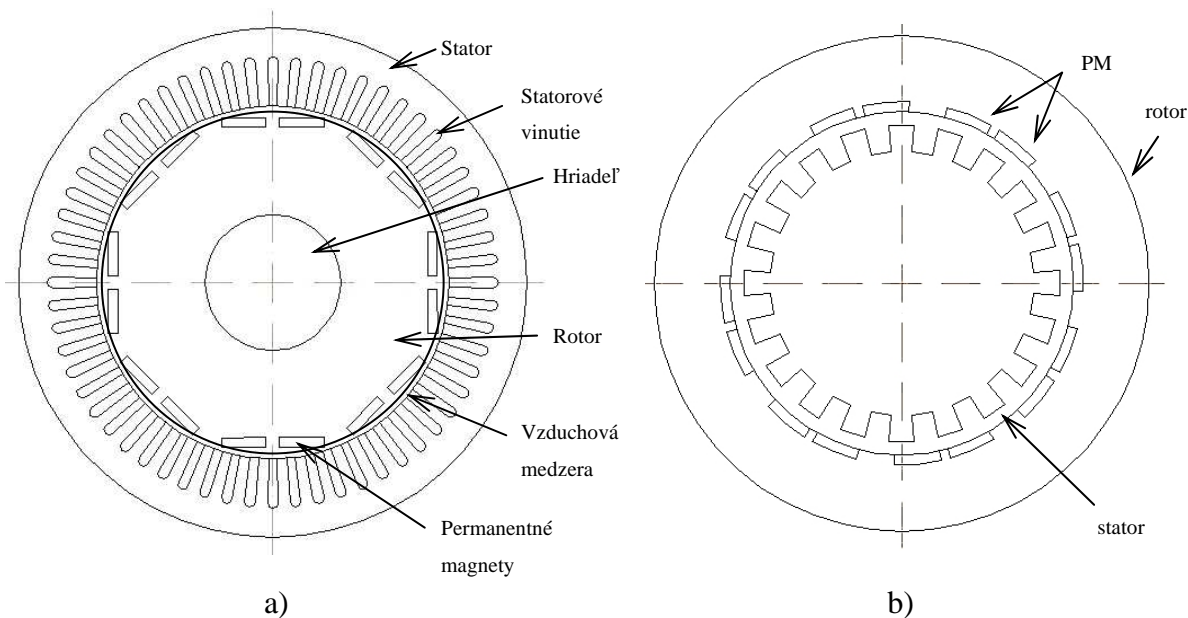
3.3 MATEMATICKÝ POPIS SYNCHRÓNNEHO MOTORA

V súčasnosti nachádzajú v praxi synchronne motory s permanentnými magnetmi (PMSM) veľké uplatnenie [8], [9]. Súvisí to hlavne s technológiou nových zliatin pre výrobu permanentných magnetov. Indukcia bežného feritu sa pohybuje okolo 0,3 – 0,4 T, pričom magnetická indukcia týchto špeciálnych zliatin v rozmedzí 0,8 – 1,2 T. Pretože sú zdrojom veľkého magnetického toku, umožňujú zmenšiť

rozmery motora a použiť ich v rôznych aplikačných oblastiach. Preto sa ďalej budeme zaoberať práve týmto modelom.



Obrázok 3.8: Závislosť magnetickej indukcie vybraných materiálov na intenzite magnetického poľa [8]



Obrázok 3.9: a) Klasická konštrukcia PMSM b) Motor s vonkajším rotorom

3.3.1 Podmienky a predpoklady

Pri modeli synchrónneho motora s permanentnými magnetmi (PM) si zavedieme niektoré zjednodušujúce predpoklady:

- Motor je bez tlmiaceho vinutia
- Saturácia je zanedbateľná
- Indukované elektromagnetické pole je sínusové
- Vírivé prúdy a hysterezia sú malé a teda zanedbateľné
- Nie sú žiadne polia dynamických prúdov
- Rotor neobsahuje klieťku

A že je napájaný harmonickým napätím (3.11) a tvorený 3-fázami, ktoré sú potrebné pre vytvorenie točivého magnetického poľa.

3.3.2 Odvodenie základných vzťahov

Pre každú z fáz synchrónneho motora môžeme vyjadriť vzťah pre okamžitú hodnotu statorového napätia

$$\begin{aligned} u_a &= R_s i_a + \frac{d\psi_a}{dt} \\ u_b &= R_s i_b + \frac{d\psi_b}{dt}, \\ u_c &= R_s i_c + \frac{d\psi_c}{dt} \end{aligned} \quad (3.49)$$

kde $u_{a,b,c}$ je okamžité statorové napätie, R_s je fázový odpor statora, $i_{a,b,c}$ je okamžitý statorový prúd a $\psi_{a,b,c}$ je okamžitá hodnota statorového spráženého toku v príslušnej fáze. Aplikovaním Parkovej transformácie na napätia a magnetické toky dostávame napäťové rovnice

$$u_d = R_s i_d - \omega_e \psi_q + \frac{d\psi_d}{dt}, \quad (3.50)$$

$$u_q = R_s i_q + \omega_e \psi_d + \frac{d\psi_q}{dt}, \quad (3.51)$$

$i_{d,q}$ prúdy statora, ω_e je elektrická rýchlosť rotora a sprážené magnetické toky $\psi_{d,q}$ sú

$$\psi_d = L_d i_d + \psi_f \quad (3.52)$$

$$\psi_q = L_q i_q \quad (3.53)$$

Indexy d, q sú osi príslušných premenných a ψ_f je vzájomný sprážený tok medzi statorom a rotorom daný PM. Elektrický moment na základe magnetických tokoch a rovníc (3.52) a (3.53)

$$T_e = \frac{3}{2} P_p (\psi_d i_q - \psi_q i_d) = \frac{3}{2} P_p [\psi_f i_q + (L_d - L_q) i_d i_q], \quad (3.54)$$

kde P_p je počet párov pólů. Pri pôsobení konštantného toku (kedy sa nemení magnetická vodivosť vo vzduchovej medzere), je i_d blízke nule a elektrický moment motora je

$$T_e = \frac{3}{2} P_p \psi_f i_q = K_t i_q. \quad (3.55)$$

Na základe pohybovej rovnice a fundamentalných pravidiel pre dynamiku dostávame vzťahy

$$\sum T_e = J \frac{d\omega_m}{dt}, \quad (3.56)$$

$$T_e = T_L + K_{TL} \omega_m + J \frac{d\omega_m}{dt}. \quad (3.57)$$

T_L je moment záťaže, K_{TL} je koeficient tlmenia (pri ďalších úpravách ho budeme zanedbávať), ω_m je mechanická rýchlosť motora a J je moment zotrvačnosti

Dosadením vzťahov (3.52), (3.53) do (3.50), (3.51), a (3.55) do (3.57) získame po úpravách rovnice stavové rovnice popisujúce PMSM

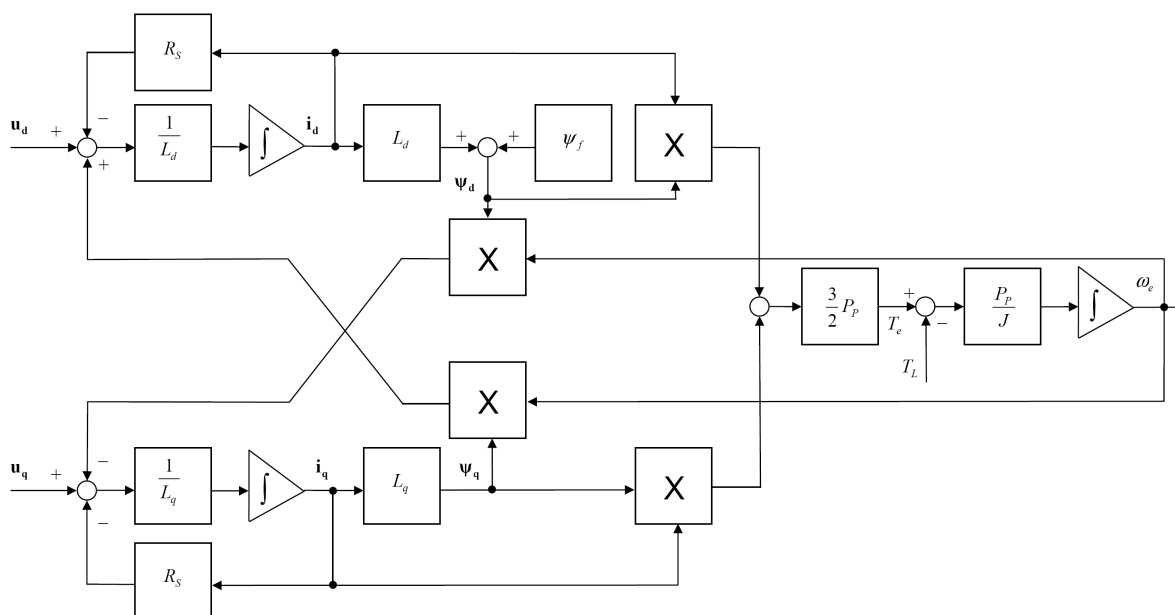
$$\frac{di_d}{dt} = \frac{u_d - R_s i_d + \omega_e L_q i_q}{L_d}, \quad (3.58)$$

$$\frac{di_q}{dt} = \frac{u_q - R_s i_q - \omega_e L_d i_d - \omega_e \psi_f}{L_q}, \quad (3.59)$$

$$\frac{d\omega_e}{dt} = \frac{P_p}{J} \left(\frac{3}{2} P_p [\psi_f i_q + (L_d - L_q) i_d i_q] - T_L \right), \quad (3.60)$$

$$\frac{d\vartheta_e}{dt} = \omega_e. \quad (3.61)$$

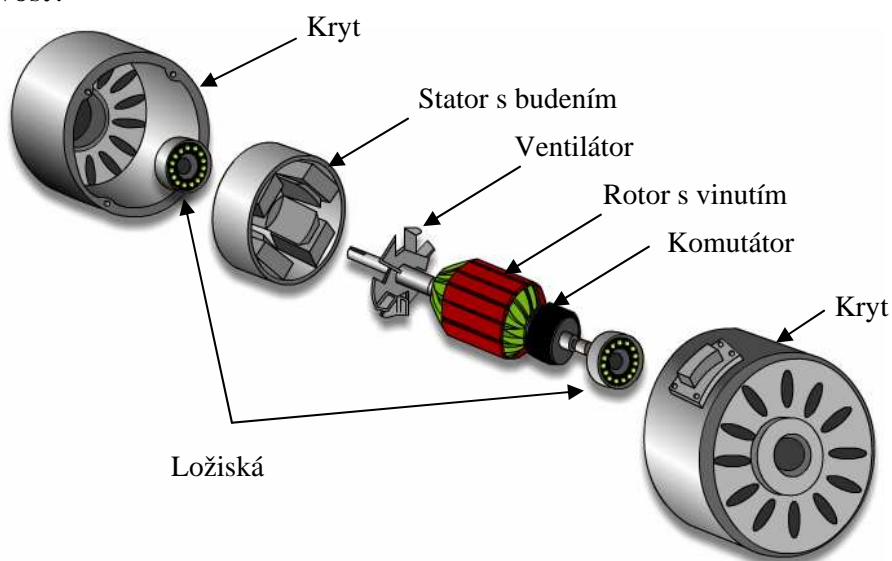
ω_e je elektrická synchronná uhlová rýchlosť ($\omega_e = \omega_s$) a ϑ_e elektrická poloha. Pripomeňme si, že prevod medzi elektrickú uhlovú rýchlosť a uhlovú rýchlosť rotora platí vzťah (3.26). Stavovým rovniciam odpovedá nasledujúca bloková schéma.



Obrázok 3.10: Bloková schéma PMSM v súradniciach d, q

3.4 MATEMATICKÝ POPIS JEDNOSMERNÉHO MOTORA

Jednosmerné motory (DC motory) neodmysliteľne patria do regulačných pohonov, a nachádzajú sa v rôznych oblastiach automatizácie. Je to hlavne pre ich relatívne nízke výrobné náklady, ľahkú reguláciu, možnosť vysokých otáčok (motory na striedavý prúd sú často obmedzované frekvenciou napájacej siete) a vysokú spoľahlivosť.



Obrázok 3.11: Základná konštrukcia DC motora [11]

Jednosmerné motory s cudzím buđením [6] majú magnetický tok kolmý na smer prúd kotvy a preto motor vyvíja stále maximálny moment. U striedavých regulačných pohonov sa dosahuje tejto vlastnosti veľmi ťažko, len za cenu zložitých regulačných obvodov. V nasledujúcich podkapitolách sa budeme zaoberať modelmi jednosmerného motora s cudzím a sériovým buđením a jednosmerného motora s permanentnými magnetmi.

3.4.1 Matematický popis DC motora s cudzím buđením

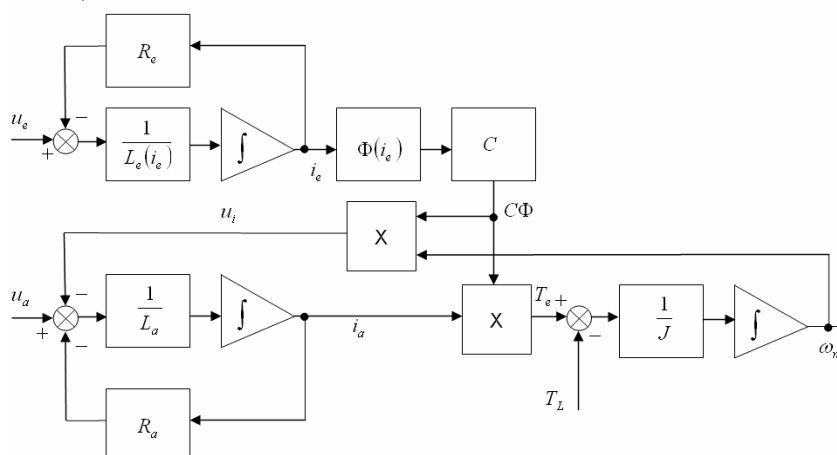
S ohľadom na model potrebný pre praktické využitie, nemusí model obsahovať všetky elektromagnetické väzby [6]. Preto zanedbáme vplyv vírivých prúdov v magnetických obvodoch, úbytok napätia na zberačoch (kefka, uhlík), rozptylový magnetický tok budiaceho vinutia, vzájomné transformačné pôsobenie jednotlivých vinutí a vplyv reaktancie kotvy. Za týchto zjednodušujúcich predpokladov môžeme DC motor s cudzím buđením popísať vzťahmi

$$\frac{di_a}{dt} = \frac{u_a - R_a i_a - C\Phi(i_e)\omega_m}{L_a}, \quad (3.62)$$

$$\frac{di_e}{dt} = \frac{u_e - R_e i_e}{L_e(i_e)}, \quad (3.63)$$

$$\frac{d\omega_m}{dt} = \frac{C\Phi(i_e)i_a - T_L}{J}, \quad (3.64)$$

kde u_a , i_a sú napätie a prúd kotvy, R_a a L_a sú odpor a indukčnosť v obvode vinutia kotvy, u_e a i_e sú napätie a prúd budiaceho vinutia, R_e a $L_e(i_e)$ sú odpor a indukčnosť budiaceho vinutia, ktorá je závislá na budiacom prúde, T_L je moment záťaže, J je moment zotrvačnosti motora a ω_m uhlová rýchlosť motora. Výraz $C\Phi(i_e)\omega_m$ predstavuje vnútorné napätie motora a výraz $C\Phi(i_e)i_a$ elektromagnetický moment motora. Závislosť budiaceho toku aj indukčnosti na budiacom prúde je daná magnetizačnou charakteristikou. Rovnice (3.61) (3.62) a (3.63) môžeme znázorniť blokovou schémou na nasledujúcom obrázku.



Obrázok 3.12: Bloková schéma DC motora s budiacim vinutím

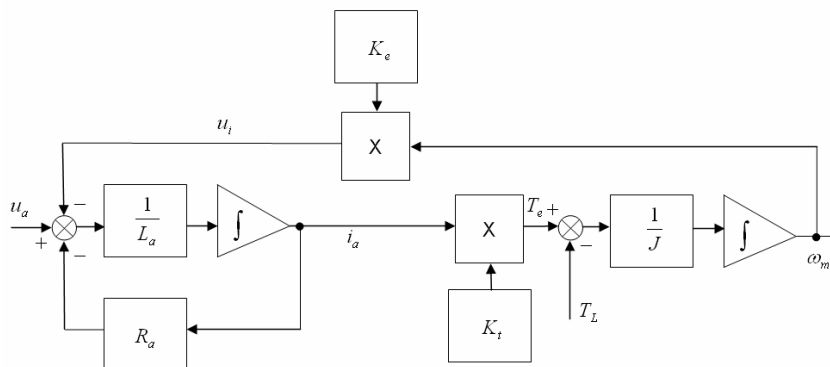
3.4.2 Matematický popis DC motora s permanentnými magnetmi

Patrí k najjednoduchším motorom pracujúcich na jednosmerný prúd [10], [12]. Stator je tvorený permanentným magnetom a rotor je tvorený elektromagnetom s pólami. Ak si uvedomíme, že permanentný magnet je zdrojom konštantného magnetického toku, tak úpravou rovníc (3.61) (3.62) a (3.63) pre DC motor s budiacim vinutím, dostávame vzťahy pre DC motor s permanentnými magnetmi

$$\frac{di_a}{dt} = \frac{u_a - R_a i_a - K_e \omega_m}{L_a}, \quad (3.65)$$

$$\frac{d\omega_m}{dt} = \frac{K_t i_a - T_L}{J}. \quad (3.66)$$

K_e je konštanta elektrického potenciálu a K_t je momentová konštanta motora.



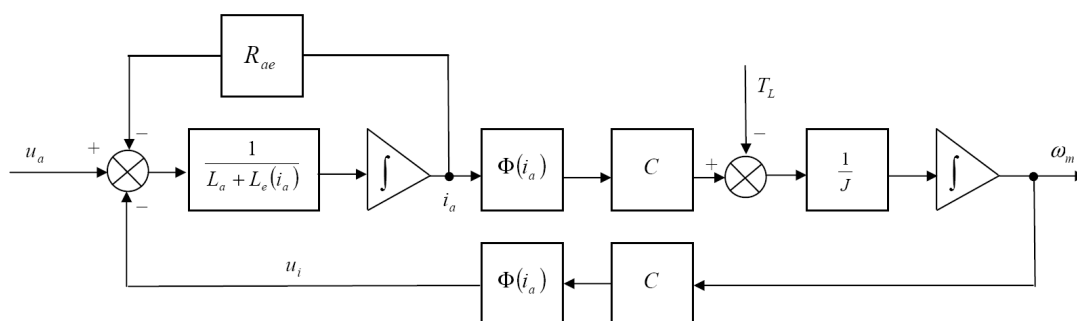
Obrázok 3.13: Bloková schéma DC motora s permanentnými magnetmi

3.4.3 Matematický popis DC motora so sériovým buzením

Budeme opäť vychádzať z rovníc pre DC motor s budiacim vinutím (3.61) (3.62) a (3.63) a s tými istými zjednodušujúcimi predpokladmi. Keďže vinutia sú zapojené do série, bude nimi tieť iba jeden prúd (i_a) a celkový odpor (R_{ae}) bude súčtom odporov jednotlivých vinutí ($R_a + R_e$). Celková indukčnosť je daná súčtom indukčnosti vinutia kotvy L_a a indukčnosťou budiaceho vinutia $L_e(i_a)$, ktoré je závislá na prúde kotvy. Výsledné vzťahy pre matematický model DC motora so sériovým buzením sú

$$\frac{di_a}{dt} = \frac{u_a - R_{ae}i_a - C\Phi(i_a)\omega_m}{L_a + L_e(i_a)}, \quad (3.67)$$

$$\frac{d\omega_m}{dt} = \frac{C\Phi(i_a)i_a - T_L}{J}. \quad (3.68)$$



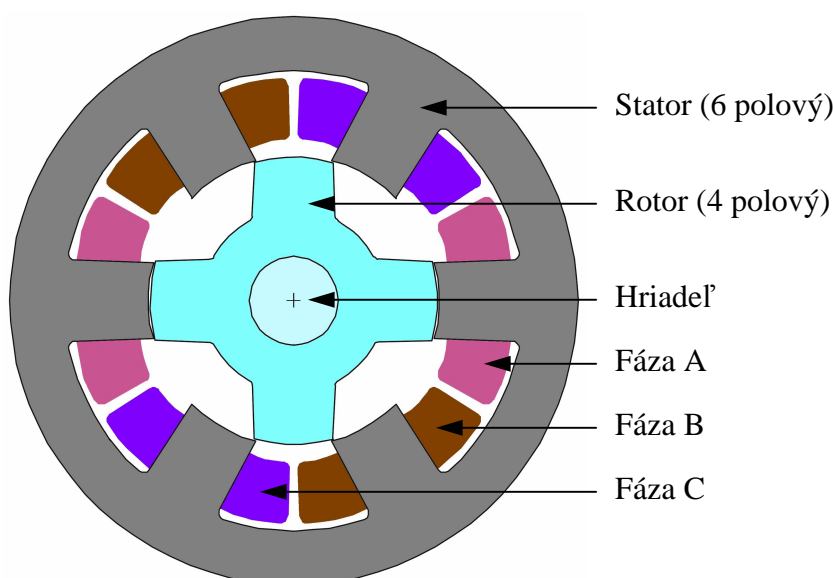
Obrázok 3.14: Bloková schéma DC motora so sériovým vinutím

3.5 SPÍNANÝ RELUKTANČNÝ MOTOR

Spínaný reluktančný motor (SRM) [13], [14], [15] je tvorený vyniknutými pólmi a to ako rotore tak aj statore, avšak s rozdielnym počtom. Počet je volený tak, aby sa minimalizoval počet spínacích prvkov. Počet statorových pólov určuje počet rotorových pólov, tak aby sa dosiahlo optimálneho točivého momentu (6/4, 12/8). Stator je najčastejšie tvorený trojfázovým vinutím okolo pólových nastavcoch, ako je znázornené na obrázku 3.15. Pri buzení jednou fázou sa rotor pootočí do energeticky najúspornejšej konfigurácií (maximálna indukčnosť) a tým sa rotor dostane do zarovnanej polohy s danou fázou (na obrázku túto situáciu reprezentuje fáza A). Keďže je rozdielny počet pólov rotora, nemôže nastať situácia, aby boli zarovnané

všetky póly rotora s fázami statora. Preto zopnutím ďalšej fázy, ktorá je nezarovnaná (indukčnosť je minimálna) sa bude proces opakovať a rotor sa zasa pootočí tak, aby magnetická cesta dosahovala najmenšieho odporu. Motor je teda budený sekvenciou pulzov pre každú fázu tak, aby sa neustále vytváral točivý moment.

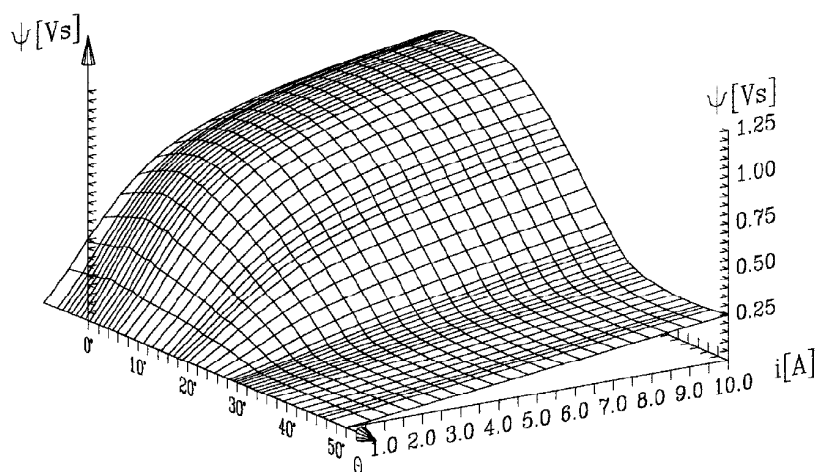
Výhodou tohto motora je jeho jednoduchá konštrukcia, nízka cena, značná nezávislosť jeho vlastností na teplote a dosahuje pomerne vysoké otáčky. Nevýhodou je jeho hlučnosť, zvlnený moment a hlavne nutnosť vedieť okamžitú polohu rotora pre správne spínanie jednotlivých fáz a teda nutnosť čidla, alebo estimátoru.



Obrázok 3.15: SRM s 6 statorovými pólmi a 4 rotorovými pólmi

3.5.1 Matematický popis SRM

Z princípu SRM je zrejmé, že celý model bude značne nelineárny a to preto, že magnetizačná charakteristika ψ bude závislá ako na polohe θ tak aj na prúde i . Ilustračná magnetizačná charakteristika je na nasledujúcom obrázku. Vplyv prúdu fázy je najviac zrejmý v oblasti zarovnanej polohy ($\theta = 0$), kedy sa prejaví saturačný efekt.



Obrázok 3.16: Magnetizačná charakteristika SR motora [15]

Na obrázku 3.17 je elektrická schéma ekvivalentu jednej fázy. Aplikovaním Kirchhoffových zákonov a za predpokladu, že sa jednotlivé fázy neovplyvňujú dostávame vzťah

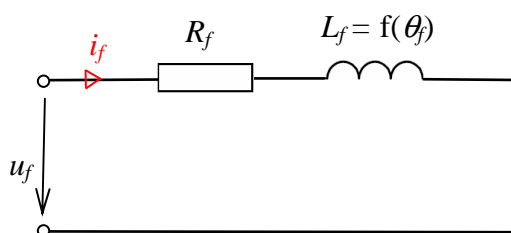
$$u_f(t) = R_f i_f(t) + u_{Lf}(t). \quad (3.69)$$

u_f je napájacie napätie fázy, R_f a i_f a je odpor a prúd fázy, u_{Lf} je indukované napätie, ktoré je definované spráženým magnetickým tokom, ktorý je funkciou prúdu i_f a polohy θ

$$u_{Lf}(t) = \frac{d\Psi_f(i_f, \theta_f)}{dt} = \frac{\partial \Psi_f(i_f, \theta_f)}{\partial i_f} \cdot \frac{di_f}{dt} + \frac{\partial \Psi_f(i_f, \theta_f)}{\partial \theta_f} \cdot \frac{d\theta_f}{dt}. \quad (3.70)$$

Po dosadení (3.70) do (3.69) a nahradení parciálnej derivácie polohy podľa času rýchlosťou obdržíme rovnicu

$$u_f(t) = R_f i_f(t) + \frac{\partial \Psi_f(i_f, \theta_f)}{\partial i_f} \cdot \frac{di_f}{dt} + \frac{\partial \Psi_f(i_f, \theta_f)}{\partial \theta_f} \cdot \omega_m. \quad (3.71)$$



Obrázok 3.17: Náhradná elektrická schéma jednej fázy SRM

Moment motora generovaný jednou fázou na základe elektromechanickej transformácie energie

$$T_{ef} = \int_0^{i_f} \frac{\partial \Psi_f(i_f, \theta_f)}{\partial \theta_f} di \quad (3.72)$$

a celkový moment motora je potom pre n fáz

$$T_e = \sum_{j=1}^{j=n} T_{ef} = \int_0^{i_a} \frac{\partial \Psi_a(i_a, \theta_a)}{\partial \theta_a} di + \int_0^{i_b} \frac{\partial \Psi_b(i_b, \theta_b)}{\partial \theta_b} di + \int_0^{i_c} \frac{\partial \Psi_c(i_c, \theta_c)}{\partial \theta_c} di. \quad (3.73)$$

Stavové rovnice odvodené z rovníc (3.71), (3.72) a (3.73), a popisáním dynamiky pohybovou rovnicou sú potom

$$\frac{di_a}{dt} = \frac{1}{\frac{\partial \Psi_a(i_a, \theta_a)}{\partial i_a}} \cdot \left(u_a - R_a i_a - \frac{\partial \Psi_a(i_a, \theta_a)}{\partial \theta_a} \cdot \omega_m \right), \quad (3.74)$$

$$\frac{di_b}{dt} = \frac{1}{\frac{\partial \Psi_b(i_b, \theta_b)}{\partial i_b}} \cdot \left(u_b - R_b i_b - \frac{\partial \Psi_b(i_b, \theta_b)}{\partial \theta_b} \cdot \omega_m \right), \quad (3.75)$$

$$\frac{di_c}{dt} = \frac{1}{\frac{\partial \Psi_c(i_c, \theta_c)}{\partial i_c}} \cdot \left(u_c - R_c i_c - \frac{\partial \Psi_c(i_c, \theta_c)}{\partial \theta_c} \cdot \omega_m \right), \quad (3.76)$$

$$\frac{d\omega_m}{dt} = \frac{1}{J} \cdot \left(\sum_{j=1}^{j=n} \int_0^{i_f} \frac{\partial \Psi_f(i_f, \theta_f)}{\partial \theta_f} di - T_L \right), \quad (3.77)$$

$$\frac{d\theta}{dt} = \omega_m. \quad (3.78)$$

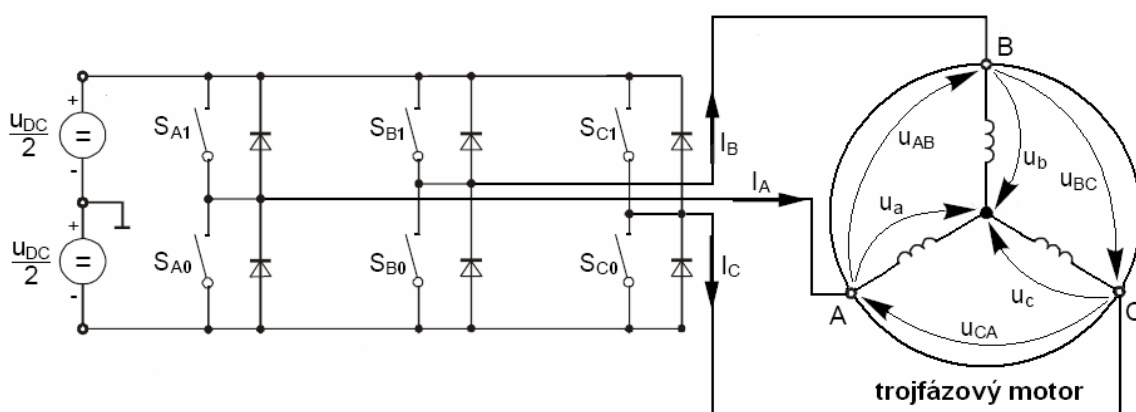
3.6 ALGORITHMUS SPACE VECTOR MODULATION

Space Vector Modulation (SVM) je algoritmus používaný pre riadenie motorov pracujúcich na striedavom prúde, primárne trojfázových. Požadované napájacie napätie (amplitúda a kmitočet) pre motor popisuje komplexor. Ten sa na základe použitej výkonovej elektroniky rozdelí na dielčie vektory, ktoré riadia pulzne šírkovú moduláciu (PWM). Rýchlym spínaním výkonovej elektroniky je možné docieľať, že prúdy sa budú podobat' harmonickým - čo je spravidla požadované,

pretože indukčnosť motora pôsobí ako filter. SVM je teda elegantná technika ako prepojiť vektorové riadenie a PWM.

3.6.1 Princíp Space Vector Modulation

Pre vytvorenie modelu SVM bolo použitý princíp podľa [18], ktorá môže transformovať komplexor napätia priamo zo súradného systému $\alpha\beta$ do PWM signálov. PWM pomerom doby zopnutia a rozopnutia vhodnej kombinácií spínačov výkonovej elektroniky, napája motor napätím zo zdroja konštantného napätia, ako je naznačené na obr. 3.18.



Obrázok 3.18: Princíp výkonovej časti

Výkonová elektronika na obrázku zobrazuje typickú trojfázovú topológiu, ktorá pozostáva zo šiestich spínacích prvkov. V každej fáze (A, B, C) je dvojica spínacích prvkov (horný $\rightarrow 1$ a dolný $\rightarrow 0$), ktorá pracujúcich v komplementárnom zapojení, tj. vždy je zopnutý len jeden prvok z dvojice. Preto môžeme zaviesť vektor spínania $[a \ b \ c]^T$ majúci hodnoty iba 0 alebo 1, kedy 1 vyjadruje zopnutý horný spínací prvok a 0 zopnutý dolný spínací prvok. Tým môžeme vyjadriť vzťah medzi vektorom spínania a napätím medzi jednotlivými fázami výrazom

$$\begin{bmatrix} U_{AB} \\ U_{BC} \\ U_{CA} \end{bmatrix} = U_{DC} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \quad (3.79)$$

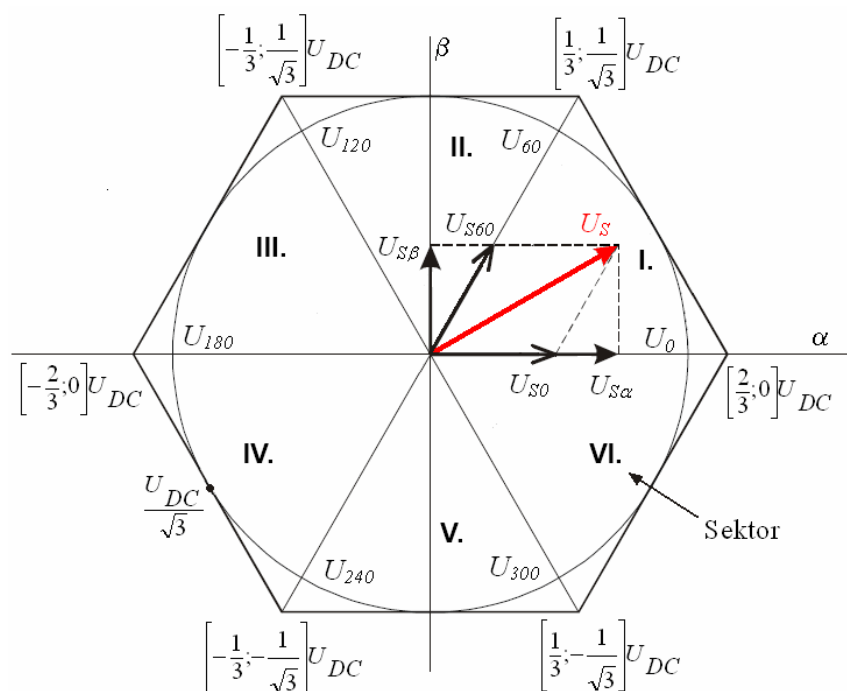
Pre ideálny symetrický motor na základe vzťahov medzi fázami dostávame

$$\begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix} = \frac{U_{DC}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (3.80)$$

a kombinácie vektoru spínania s pomerom napätia fáze k DC napätiu zdroja sú v nasledujúcej tabuľke

Tabuľka 3.1: Generované napätia v závislosti na kombinácii vektoru spínania

Vektor	a	b	c	$\frac{U_a}{U_{DC}}$	$\frac{U_b}{U_{DC}}$	$\frac{U_c}{U_{DC}}$
O ₀₀₀	0	0	0	0	0	0
U ₀	1	0	0	2/3	-1/3	-1/3
U ₆₀	1	1	0	1/3	1/3	-2/3
U ₁₂₀	0	1	0	-1/3	2/3	-1/3
U ₁₈₀	0	1	1	-2/3	1/3	1/3
U ₂₄₀	0	0	1	-1/3	-1/3	2/3
U ₃₀₀	1	0	1	1/3	-2/3	1/3
O ₁₁₁	1	1	1	0	0	0



Obrázok 3.19: Princíp SVM

Na obrázku 3.19 je znázornený princíp SVM. Vektor požadovaného napätia v α, β súradniciach sa rozloží pomocou vektorov z tab. 3.1. tak, aby stredná hodnota výstupného vektoru napätia z PWM bola zhodné s požadovaným vektorom. To sa dosiahne správnym časovým pomerom jednotlivých aktívnych vektorov počas jednej periódy PWM. Je viacero možností ako voliť jednotlivé aktívne vektory a ich doby. Jedným z najjednoduchších a najefektívnejších spôsoboch je požadovaný vektor zostaviť pomocou nenulových vektorov (U_{xxx}) definujúcich - ohraničujúcich sektor (U_x a $U_{x\mp 60}$), v ktorom sa nachádza a nulových vektorov (O_{xxx}).

Na základe takto definované podmienky môžeme prepísať do vzťahov

$$T_{PWM} U_{s[\alpha,\beta]} = T_1 U_x + T_2 U_{x\mp 60} + T_0 O_{xxx}, \quad (3.81)$$

$$T_{PWM} = T_1 + T_2 + T_0. \quad (3.82)$$

Po rozložení požadovaného vektoru $U_{s[\alpha,\beta]}$ pomocou základných nenulových vektorov (U_{sx} a $U_{sx\mp 60}$) sa rovnica (3.81) rozpadne do dvoch rovníc, tj. jedna rovnica pre každý nenulový vektor. Z týchto rovníc po vyjadrení jednotlivých časových intervalov dostávame

$$T_1 = \frac{|U_{sx}|}{|U_x|} T_{PWM}, \quad (3.83)$$

$$T_2 = \frac{|U_{sx\mp 60}|}{|U_{x\mp 60}|} T_{PWM}, \quad (3.84)$$

$$T_0 = T_{PWM} - T_1 - T_2. \quad (3.85)$$

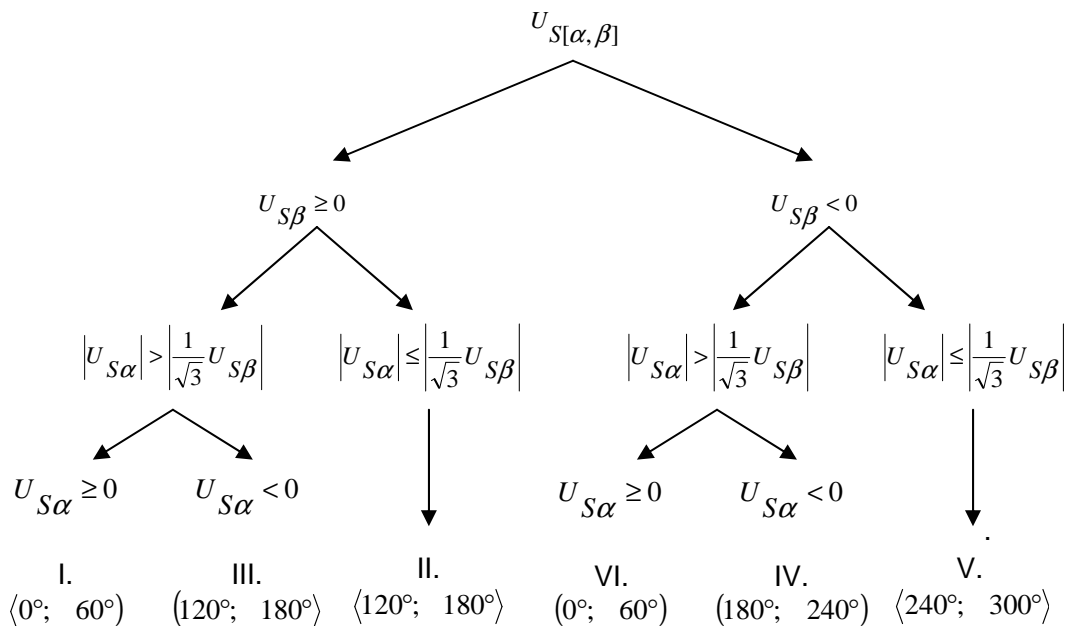
3.6.2 Algoritmus pre aplikovanie SVM

Celý algoritmus pozostáva z troch krokov a je veľmi jednoduchý pre implementovanie do modelu, alebo na mikroprocesore.

- 1) Identifikácia sektoru
- 2) Dekompozícia napäťového vektoru
- 3) Výpočet zložiek PWM

3.6.2.1 Identifikácia sektoru

Identifikácia sektoru, kde sa nachádza vektor sa dá zistiť priamo zo zložiek α , β vektoru. Popis identifikácie je na zobrazený na strome identifikácie sektoru, ktorý bol odvodený na základe triangulárnej geometrie.



Obrázok 3.20: Strom identifikácie sektoru

3.6.2.2 Dekompozícia napät'ového vektoru

Na základe identifikovaného sektoru sa vektor $U_{S[\alpha, \beta]}$ rozloží do základných vektorov ohraničujúcich sektor na $U_{S\alpha}$ a $U_{S\alpha \mp 60}$ podľa vzťahov

Sektor I.

$$U_{S0} = |U_{S\alpha}| - \left| \frac{1}{\sqrt{3}} U_{S\beta} \right| \quad U_{S60} = \left| \frac{2}{\sqrt{3}} U_{S\beta} \right| \quad (3.86)$$

Sektor II.

$$U_{S60} = U_{S\alpha} + \left| \frac{1}{\sqrt{3}} U_{S\beta} \right| \quad U_{S120} = -U_{S\alpha} + \left| \frac{1}{\sqrt{3}} U_{S\beta} \right| \quad (3.87)$$

Sektor III.

$$U_{S120} = \left| \frac{2}{\sqrt{3}} U_{S\beta} \right| \quad U_{S180} = |U_{S\alpha}| - \left| \frac{1}{\sqrt{3}} U_{S\beta} \right| \quad (3.88)$$

Sektor IV.

$$U_{s180} = |U_{s\alpha}| - \left| \frac{1}{\sqrt{3}} U_{s\beta} \right| \quad U_{s240} = \left| \frac{2}{\sqrt{3}} U_{s\beta} \right| \quad (3.89)$$

Sektor V.

$$U_{s240} = -U_{s\alpha} + \left| \frac{1}{\sqrt{3}} U_{s\beta} \right| \quad U_{s300} = U_{s\alpha} + \left| \frac{1}{\sqrt{3}} U_{s\beta} \right| \quad (3.90)$$

Sektor VI.

$$U_{s300} = \left| \frac{2}{\sqrt{3}} U_{s\beta} \right| \quad U_{s0} = |U_{s\alpha}| - \left| \frac{1}{\sqrt{3}} U_{s\beta} \right| \quad (3.91)$$

3.6.2.3 Výpočet zložiek PWM

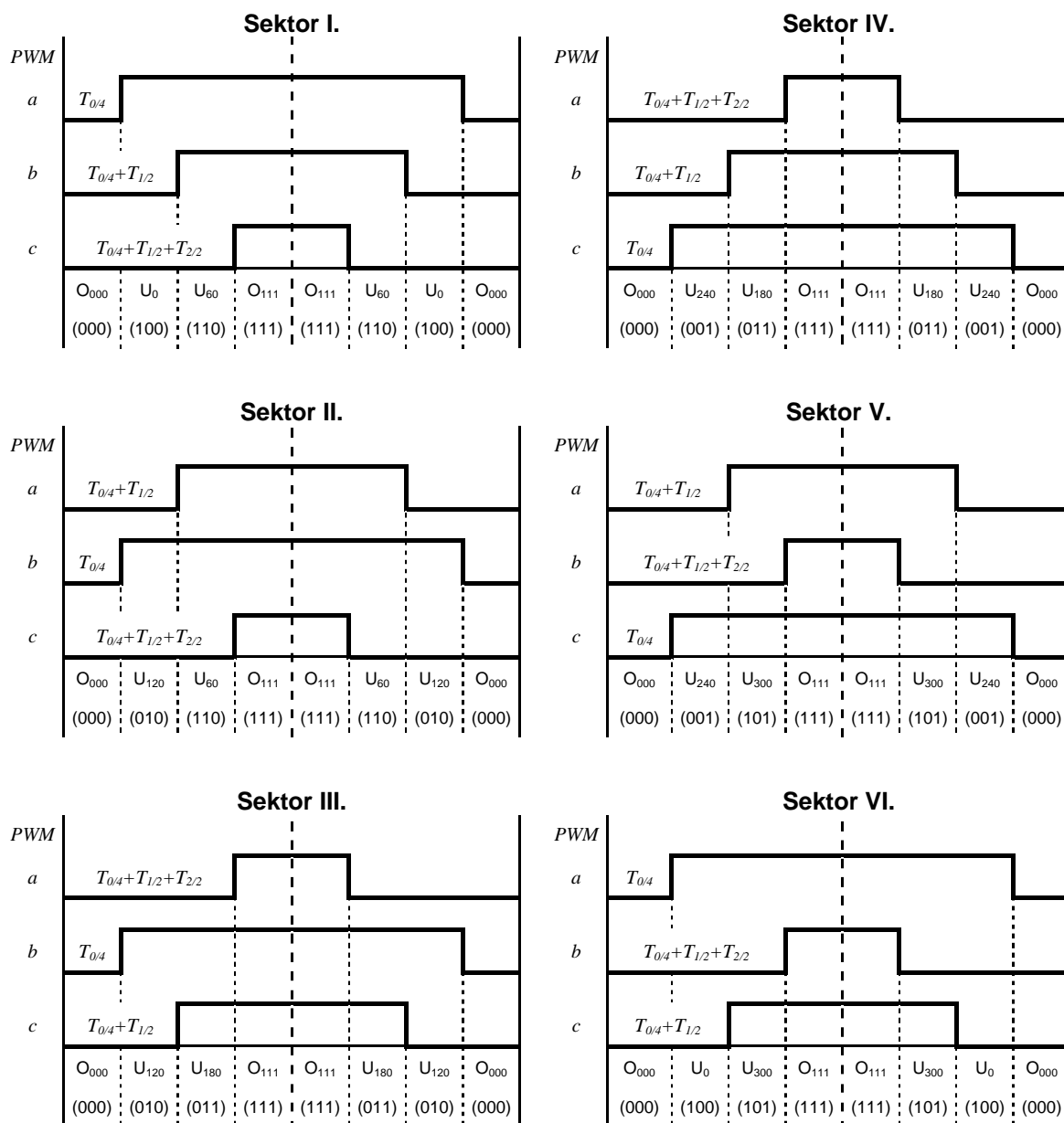
Pre výpočet jednotlivých zložiek PWM použijeme centrovanú štruktúru PWM (obr. 3.21). Táto štruktúra je vhodná, aj vzhľadom k tomu, že umožňuje jednoduché synchronizované meranie veličín (prúdu) práve týmto centrovaným systémom.

Rovnice popisujúce časové intervaly, ktoré základné vektory sú kedy aktívne, transformujeme do nových vzťahov, ktoré namiesto času udávajú počet impulzov počítadla. Počítadlo je realizované ako vzostupne a zostupne. Počet impulzov počas jednej periódy T_{PWM} je $2 \times Modulus$, kde *Modulus* je maximálna hodnota počítadla (obr. 3.22). Z rovníc (3.83), (3.84) a (3.85) a na základe obrázku 3.22, môžeme napísať vzťahy

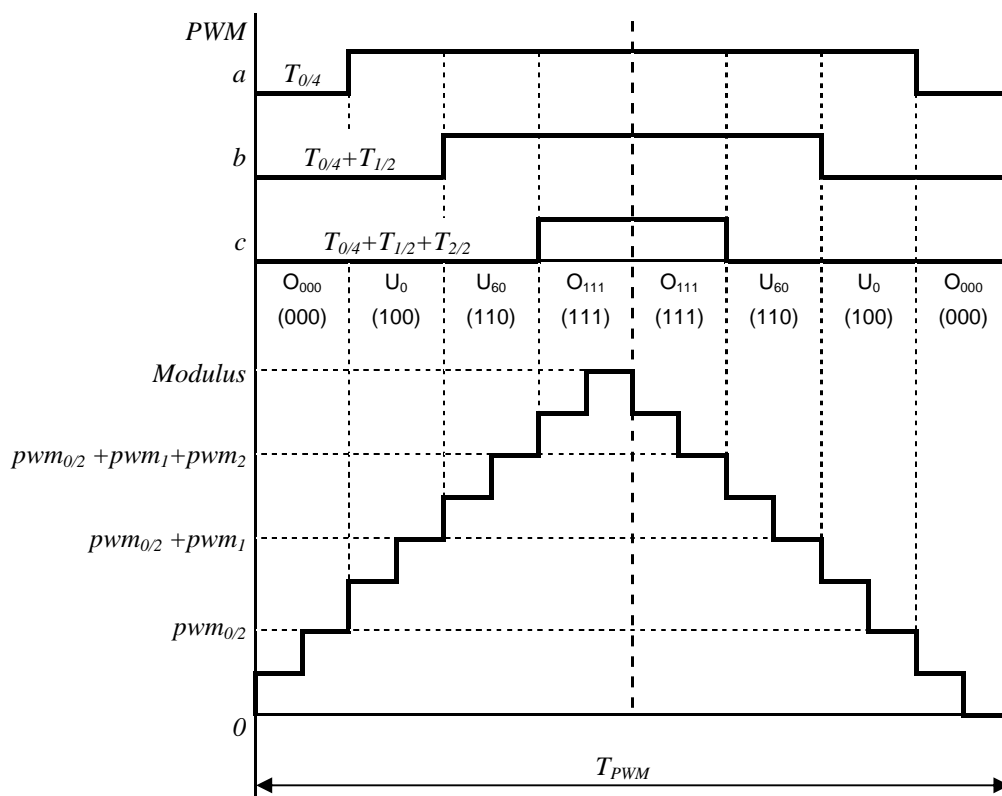
$$pwm_1 = \frac{|U_{sx}|}{|U_x|} Modulus, \quad (3.92)$$

$$pwm_2 = \frac{|U_{sx \mp 60}|}{|U_{x \mp 50}|} Modulus, \quad (3.93)$$

$$pwm_0 = Modulus - pwm_1 - pwm_2. \quad (3.94)$$



Obrázok 3.21: Centrovaná štruktúra PWM



Obrázok 3.22: Detail sektoru I.

Z uvedených vzťahov (3.92), (3.93) a (3.94) a z obrázka 3.21 a 3.22 vyplýva vzťah medzi riadiacimi impulzmi (A,B,C) pre výkonové spínače a hodnotou počítadla. Ak hodnota počítadla presiahne úroveň p_{wm} , tak sa aktivuje jemu odpovedajúci vektor, tj. výkonové spínacie prvky.

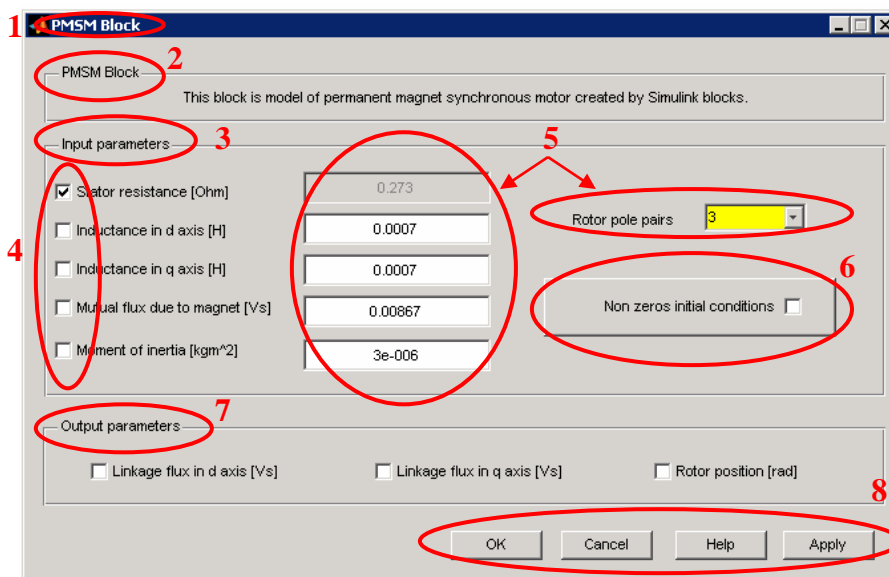
4. TVORBA MODELOV

4.1 NÁVRH UŽIVATELSKÉHO ROZHRAŇIA PRE MODELY

Všetky modely motorov boli zostavované s rovnakým postupom, ktorý sa líšil, iba variantov a samotnými vlastnosťami daného modelu. Preto princíp činnosti a spôsob implementácia popísaný v nasledujúcich kapitolách, platí pre všetky modely. Prípadne rozdiely a špecifika budú objasnené pri konkrétnom modeli.

4.1.1 Grafická časť

Z kapitoly 2 vyplýva, že pre užívateľské rozhranie modelov motorov bude vhodné použiť GUI programu Matlab, ktorá dáva viac priestoru pre vytvorenie kompaktného rozhrania. Na obrázku 4.1 je ukážka užívateľského rozhrania, ktorá predstavuje koncept pre všetky užívateľské rozhrania modelov motorov. Jednotlivé časti si teraz priblížime.



Obrázok 4.1: Užívateľské rozhrania pre model motora – PMSM

1 Názov bloku

Názov GUI, odpovedá názvu bloku, ktorý ho vyvolal. Užívateľ má prehľad ktorému bloku patrí GUI, čo je dôležité, ak má súčasne otvorené viaceré rozhrania.

2 Názov modelu

Hneď na vrchu okna sa nachádza názov modelu – jeho skratka so stručným popisom.

3 Pole so vstupnými parametrami

Ohraničuje skupinu prvkov, ktoré predstavujú základne parametre pre model.

4 Zaškrtavacie políčka

Každému parametru, ktorý môže byť použitý ako vstup pre model, je priradené zaškrtavacie políčko. Jeho označením sa vytvorí nový vstup a parameter, ktorý je zadaný číselne (5) sa ignoruje a zablokuje sa jeho editácia.

5 Hodnoty parametrov

Konkrétne hodnoty parametrov, vyplní užívateľ do týchto editačných okienok (vľavo). Prípadne si môže konkrétnu hodnotu vybrať zo zoznamu (vpravo).

6 Tlačidlo pre nastavenie počiatočných podmienok

Predpokladá sa, že užívateľ nebude meniť hodnoty počiatočných podmienok tak často, ako hodnoty ostatných parametroch. Preto, aby nenavyšovali rozmer a prehľadnosť okna bolo im priradené samostatné okno (obr. 4.3). Prístup k ich nastaveniu je práve cez toto tlačidlo. Tlačidlo je doplnené o zaškrtavacie políčko, ktoré signalizuje, či počiatočné stavové hodnoty sú, alebo nie sú nulové. V prípade prázdneho políčka sa použijú nulové počiatočné podmienky. Inak sa použijú hodnoty zadané v okne počiatočných podmienok.

7 Pole s výstupnými parametrami

Ohraničuje veličiny, pre ktoré môže užívateľ priradiť výstupy z modelu. Označením parametru, sa pre danú veličinu vytvorí výstup.

8 Potvrdzovacie tlačidlá

Zostava tlačidiel, typická pre mnohé rozhrania.

OK – uloží nastavenia a ukončí dialógové okno.

Cancel – ukončí dialógové okno bez uloženia zmien užívateľom

Help – otvorí okno s nápodnou k modelu a ovládaniu užívateľského rozhrania

Apply – uloží zmeny avšak ponechá okno otvorené pre prípadne ďalšie zmeny

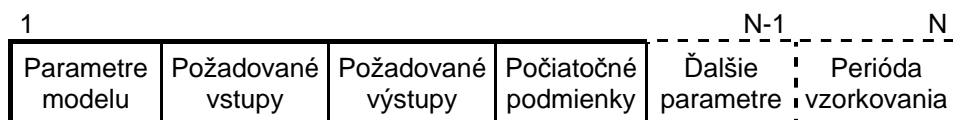
Pre niektoré modely motorov bolo vhodné definovať závislosť veličiny na inej veličine (nelineárne funkcie). Preto sú niektoré rozhrania doplnené grafické znázornenia týchto závislostí, alebo o ďalšie tlačidlá, cez ktoré sa tieto závislosti nastavujú.

4.1.2 Funkcie a správa dát pre GUI

Ako prvé pri spustení GUI sa zistí, ktorý blok ju vyvolal a kde sa nachádza. To sa uloží do premenných *BlockPath* a *BlockName* (kap. 2.6), aby sme vedeli z ktorého bloku sa majú parametre – dáta načítať a kde sa majú uložiť. Názov bloku nastaví aj ako názov GUI, čím sa evidentne sparuje s príslušným blokom, aby bolo aj pre užívateľa zrejme, ku ktorému bloku patrí GUI, hlavne ak má súčasne otvorené viaceré rozhrania (obr. 4.1 – 1). Ukážka kódu, ktorý to realizuje:

```
handles.BlockPath = gcs; % získaj cestu bloku
handles.BlockName = get_param(gcf, 'Name'); % získaj názov bloku
set(handles.figure, 'Name', [handles.BlockName]); % nastav názov GUI
```

Následne sa načítajú všetky parametre, ktoré boli nastavené v modeli a uložia sa do štruktúry platných dát *handles.ValidData*. Parametre sa pre „blokovú“ variantu modelu načítajú priamo z jednotlivých blokov subsystémov. Pre modely vytvorených cez *S-function* sa načíta reťazec znakov (string) z poľa *Parameters*. Reťazec sa rozdelí na jednotlivé parametre, ktoré sa uložia ako číselné hodnoty. Na nasledujúcom obrázku je ukážka, v akom poradí sa skladajú jednotlivé parametre do štruktúry platných dát. Pre modely v diskkrétnej variante sa pridá na koniec štruktúry hodnota periódy vzorkovania.



Obrázok 4.2: Usporiadanie parametrov pre štruktúru platných dát

Štruktúra platných dát bola vytvorená, aby vždy boli zachované a jednoznačné platné dáta, hlavne v prípade, kedy užívateľ modifikuje parametre, ale urobené zmeny sa nepoužijú (tlačidlo Cancel). Načítané dáta sa uložia do jednotlivých komponent grafického rozhrania a tým aj sprístupnené k modifikácií pre užívateľa. Dáta sa uložia do štruktúry platných dát pri kliknutí ľavým tlačidlom myši na tlačidlo Apply, alebo OK

4.1.2.1 Callback Apply

Po kliknutí na tlačidlo Apply sa načítajú dáta z jednotlivých komponent a uložia do štruktúry platných dát *handles.ValidData*. Taktiež sa tieto dáta nastaví

ako parametre modelu. Model sa aktualizuje z hľadiska vstupov a výstupov. Preto sa hneď priradia názvy pre vstupy a výstupy bloku a celkový dizajn bloku. Sú to príkazy, ktoré sa uložia ako reťazec do parametru *MaskDisplay*, definujúcej vzhľad ikony bloku:

```
set_param([handles.BlockPath '/' handles.BlockName],...
'MaskDisplay', StringOfCommands)); % Nastav vzhľad ikony
```

4.1.2.2 Callback OK

Funkcia pre odozvu na kliknutie tlačidlo OK je obdobná ako pre Apply. V odozve sa spustí *Callback* Apply. Otestuje sa úspešnosť uloženia dát pre model, ak je to potrebné (napr. existencia požadovaných súborov dát) a ukončí sa užívateľské rozhranie.

4.1.2.3 Callback Cancel

Ukončí sa užívateľské rozhranie. Dáta sa neaktualizujú a použije sa posledná platná konfigurácia parametrov.

4.1.2.4 Callback Help

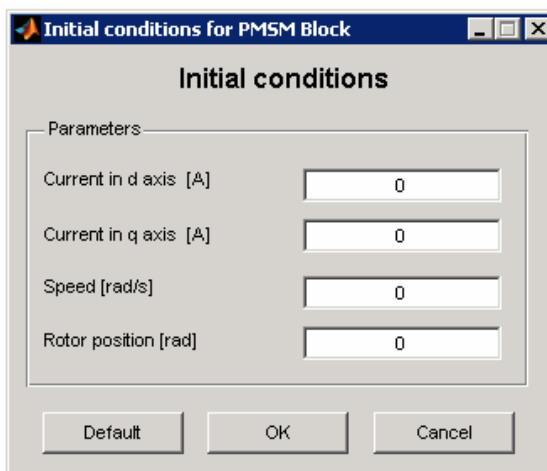
Otvorí sa nápoveda pre daný model v okne – prehliadači nápovedy pre program Matlab. Nápoveda je vytvorená v jazyku html a uložená v prečišku Help. Ukážka pre spustenie nápovedy pre model asynchrónneho motora.

```
web([PathLibrary '\Help\ASM_help.html'], '-helpbrowser'); % Nápoveda
```

4.1.2.5 Callback Initial conditions

Po kliknutí na tlačidlo Non zeros initial conditions (obr. 4.1 – 6) sa spustí ďalšie užívateľské rozhranie GUI, pričom sa v tejto funkcii predá do rozhrania štruktúra *handles* pôvodného GUI. V rozhraní sú len komponenty reprezentujúce počiatočné podmienky modelu, ako je vidieť na obrázku 4.3 pre PMSM .

Pre nové rozhranie sa vytvorí aj nová štruktúra *handles*. Do nej je pridaná štruktúra *handles* z rozhrania, ktoré spustilo toto rozhranie. Z pridanej štruktúry sa získajú dáta pre nastavenia hodnôt komponent počiatočných podmienok a aj pre názov rozhrania.



Obrázok 4.3: Uživatelské rozhranie pre nastavenie počiatkových podmienok

Rozhranie má tri tlačidlá Default, OK a Cancel. Predpokladá sa, že najčastejšie a teda aj východzie počiatkové podmienky budú nulové. Kliknutím na tlačidlo Default sa všetky hodnoty nastaví na nulu. Tlačidlom OK uložia vykonané zmeny a ukončí sa rozhranie pre nastavenie počiatkových podmienok. Tlačidlo Cancel ukončí rozhranie bez uloženie vykonaných zmien.

Uvedené funkcie *Callback* (4.1.2.1-5) sú odozvy, ktoré sú v každom rozhraní pre modely motorov. V uživatelských rozhraniach sú samozrejme použité aj ďalšie funkcie, ktoré zabezpečujú korektné fungovanie grafického rozhrania, správnu funkčnosť interpretácie dát, načítania dát pre model zo súboru apod. Ako príklad si môžeme uviesť odozvu na označenie nového vstupu pre model. Pri označení voľby pre nový vstup, sa pre daný parameter zablokuje užívateľovi modifikovať číselnú hodnotu, aby bolo zrejmé, že sa táto hodnota pre model nepoužije. A naopak pri zrušení parametru ako vstupu sa sprístupní komponenta pre číselnú hodnotu parametru. Pre komponentu statorový odpor ako nový vstup (obr. 4.1 – 4), odpovedá kód:

```
% If checkbox is on (max) -> new input for model
if (get(hObject, 'Value') == get(hObject, 'Max'))
    set(handles.EB_Stator_Resistance, 'Enable', 'off');
else
    set(handles.EB_Stator_Resistance, 'Enable', 'on');
end
```

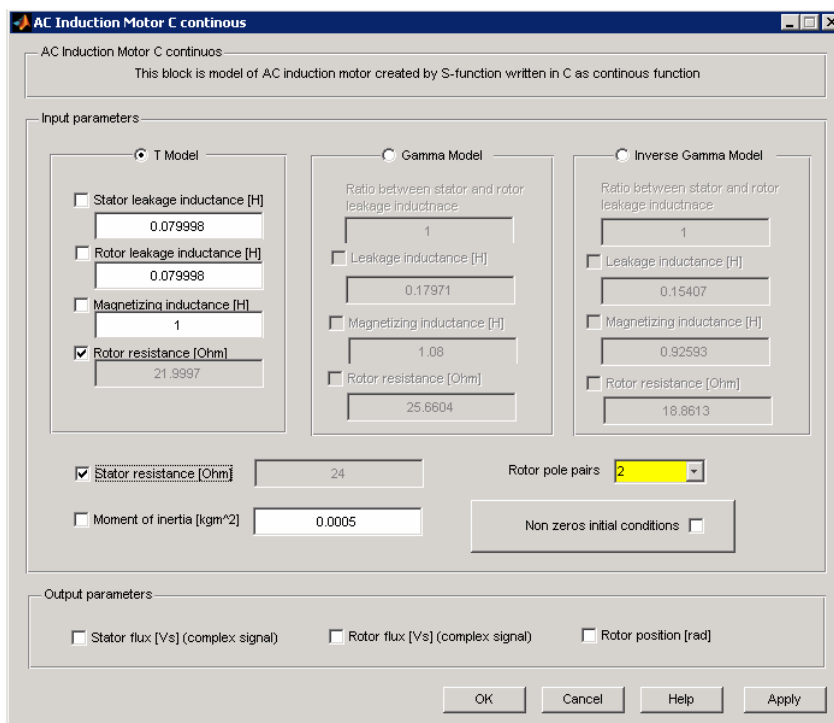
4.2 MODEL ASYNCHRÓNNEHO MOTORA

Všetky modely (ASM, PMSM, SRM atď.) boli doplnené o výstup polohy rotora motora ϑ_m .

$$\frac{d\vartheta_m}{dt} = \omega_m \quad (4.1)$$

Pre model sme použili rovnice popisujúce T-model asynchrónneho motora (3.31), (3.32), (3.33) a (4.1). Aby užívateľ mohol pracovať s Γ -modelom a inverzným Γ -modelom, pridali sa k modelu aj transformácie medzi týmito modelmi, pričom model má stále vnútornú štruktúru ako T-model, iba príslušné vstupné, výstupné signály a parametre sa prepočítavajú.

Užívateľské rozhranie sa rozšírilo o možnosť výberu typu modelu. Súčasne bolo navrhnuté tak, aby užívateľ mohol pozorovať hodnoty parametrov ostatných modelov (obr. 4.4). Vždy po zadaní hodnoty parametru v jednom type modelu sa prepočítajú a aktualizujú hodnoty komponent v ostatných typoch modelu. Tým sa užívateľ môže ľubovoľne prepínať medzi typmi modelmi a pri prepínaní nepríde o dáta, ktoré zadal v inom modeli.



Obrázok 4.4: Užívateľské rozhranie pre model ASM

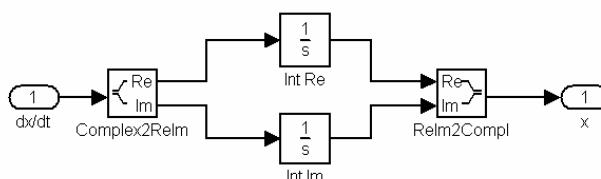
Transformácia parametrov medzi typmi modelov prebieha podľa vzťahov (3.43) pre Γ -model a (3.48) pre inverzný Γ -model. Uživateľské rozhranie pre Γ -model a inverzný Γ -model je navyše doplnené o komponentu (Ratio) vyjadrujúcu pomer medzi rozptylovými indukčnosťami $L_{S\sigma}$ a $L_{R\sigma}$, aby bolo možné správne transformovať parametre do T-modelu., ktorý má o parameter navyše oproti Γ -modelu, či a inverznom Γ -modelu. Keďže užívateľ má možnosť zvoliť namiesto použitia parametrov vstupy pre všetky typy modelov, musia transformácie parametrov byť nie len na úrovni užívateľského rozhrania, ale aj priamo v samotnom modeli. Do modelu je poslaná informácia o aký typ modelu sa jedna, aby sa zbytočne nenavýšoval počet parametrov. Keďže sa jedná o tri typy modelov, použije sa jeden parameter pre viacero vlastností podľa zvoleného typu modelu. Vymenovanie jednotlivých parametrov vstupujúcich do modelu ASM s krátkym popisom sú v tabuľke 4.1. Ak pre nieaký model nie je uvedený parameter, znamená to, že hodnota na tejto pozícii sa ignoruje.

Tabuľka 4.1: Vstupné parametre do modelu ASM

P.č.	Symbol		Popis
	T-model	Γ / Γ^{-1} -model	
1	$L_{S\sigma}$	L_M	Vstupné parametre
2	$L_{R\sigma}$	L_L	
3	L_m	-	
4	R_S		
5	R_R	R_{Rr}	
6	J		
7	P_P		
8	$InpL_{S\sigma}$	$InpL_M$	Definovanie vstupov modelu 1 = nový vstup 0 = parameter
9	$InpL_{R\sigma}$	$InpL_L$	
10	$InpL_m$	-	
11	$InpR_S$		
12	$InpR_R$	$InpR_{Rr}$	
13	$InpJ$		

P.č.	Symbol		Popis
	T-model	Γ / Γ^{-1} -model	
14	$Out\psi_S$		Definovanie výstupov modelu (1 / 0)
15	$Out\psi_R$	$Out\psi_{Rr}$	
16	$Out\vartheta_m$		
17	ψ_{Sre}		Počiatočné podmienky
18	ψ_{Sim}		
19	ψ_{Rre}		
20	ψ_{Rim}		
21	ω_m		
22	ϑ_m		
23	Type		Typ modelu
24	-	Ratio	Pomer $L_{S\sigma}/L_{R\sigma}$
25	T_S		Periódá vzorkovania

Pre implementáciu modelu ASM bolo nutné využiť komplexné premenné a signály. Práca s komplexnými veličinami je v S-function plne podporovaná. Blok integrátoru pre komplexné čísla sa nenachádza medzi základnými blokmi knižnice Matlab Simulink. Preto sme si ho „vyrobili“ zo základných blokov.



Obrázok 4.5: Ukážka integrátoru pre komplexný signál

4.3 MODEL SYNCHRÓNNEHO MOTORA S PM

Model PMSM bol implementovaný v prostredí Matlab Simulink na základe matematického popisu motora rovnicami (3.58) až (3.59) a (4.1). Ukážka užívateľského rozhrania pre tento model sa nachádza na obrázku 4.1 a ukážka rozhrania pre nastavenie počiatočných podmienok je na obrázku 4.2 V nasledujúcej tabuľke je zoznam parametrov použitých pre model PMSM s krátkym popisom.

Tabuľka 4.2: Vstupné parametre do modelu PMSM

P.č.	Symbol	Popis
1	R	Vstupné parametre
2	L_d	
3	L_q	
4	ψ_f	
5	J	
6	P_p	
7	$InpR$	Definovanie vstupov modelu 1 = nový vstup 0 = parameter
8	$InpL_d$	
9	$InpL_q$	
10	$Inp\psi_f$	
11	$InpJ$	

P.č.	Symbol	Popis
12	$Out\psi_s$	Definovanie výstupov modelu (1 / 0)
13	$Out\psi_R$	
14	$Out\vartheta_m$	
15	i_s	Počiatočné podmienky
16	i_R	
17	ω_m	
18	ϑ_m	
19	T_s	Periódá vzorkovania

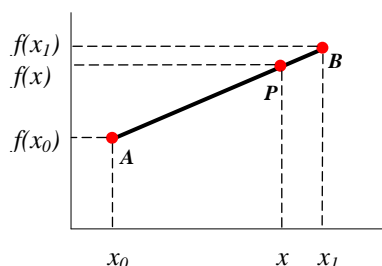
4.4 MODEL DC MOTORA

4.4.1 Model DC motora s cudzím buđením

Na základe rovníc (3.62), (3.63), (3.64) a (4.1) bol vytvorený model DC motora s cudzím buđením. Ako bolo uvedené v kapitole 3.4.1 budiaci tok $\Phi(i_e)$ aj indukčnosť $L_e(i_e)$ je závislá na budiacom prúde na základe magnetizačnej

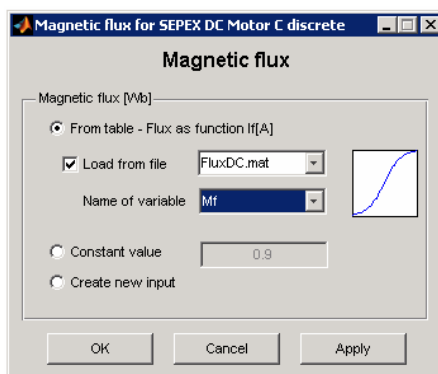
charakteristiky. Túto závislosť užívateľ môže riešiť zavedením funkcie z vyvedeného výstupu na vstup. Inou alternatívou je definovať vzťah medzi parametrom a prúdom pomocou tabuľky. Pre zistenie hodnoty, ktorá nie je presne definovaná v tabuľke, sa použije lineárna interpolácia medzi dvoma najbližšími hodnotami (A a B) znázornenej na obrázku 4.6. Vstupné hodnoty sú označené x_i (prúd) a výstupne $f_x(x_i)$ (magnetický tok, indukčnosť budiaceho vinutia). Body A a B sú z známe z tabuľky, x je aktuálna hodnota prúdu a je $f_x(x)$ hľadaná hodnota [16]

$$f_x(x) = \frac{f_x(x_1) - f_x(x_0)}{x_1 - x_0}(x - x_0) + f_x(x_0). \quad (4.2)$$

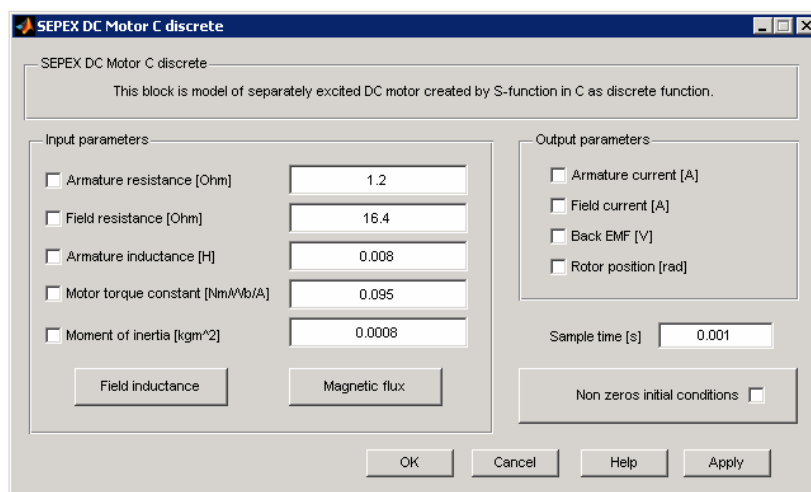


Obrázok 4.6: Lineárna interpolácia

Tabuľku je možné načítať z *Workspace*, alebo z dátového súboru pre program Matlab (*.mat), ktorý musí byť uložený v pracovnom adresári. Pri načítaní z dátového súboru sa vybraná premenná uloží do *Workspace* a model ďalej pracuje už len s premennou vo *Workspace*. Pre spätnú kontrolu správneho načítania dát slúži užívateľovi náhľad na priebeh závislosti v ikonke. Zavedenie týchto požiadavok by vnieslo neprehľadnosť do hlavného GUI. Preto do užívateľského rozhrania boli pridané tlačidlá, ktoré otvoria nové samostatné okná pre nastavenie (obr. 9.7 a 4.8).



Obrázok 4.7: Užívateľské rozhranie pre nastavenie magnetického toku



Obrázok 4.8: Uživatelské rozhranie pre model DC s cudzím buđením

Model pracuje s parametrami uvedených v tabuľke 4.3. Parametre 23 a 24 obsahujú názvy premenných vo *WorkSpace* v ktorých sú definované ich závislosti na prúde I_e pomocou tabuľky.

Tabuľka 4.3: Vstupné parametre do modelu DC motora s cudzím buđením

P.č.	Symbol	Popis
1	R_a	Vstupné parametre
2	R_e	
3	L_a	
4	L_e	
5	Φ	
6	C	
7	J	
8	$InpR_a$	Definovanie vstupov modelu 0 = parameter 1 = nový vstup $InpL_e$ a $Inp\Phi$: 0 = načítanie z tab. 1 = parameter 2 = nový vstup
9	$InpR_e$	
10	$InpL_a$	
11	$InpL_e$	
12	$Inp\Phi$	
13	$InpC$	
14	$InpJ$	

P.č.	Symbol	Popis
15	$OutI_a$	Definovanie výstupov modelu (1 / 0)
16	$OutI_e$	
17	$OutU_i$	
18	ϑ_m	
19	I_a	Počiatočné podmienky
20	I_e	
21	ω_m	
22	ϑ_m	
23	$TableL_e$	Názov premennej (tab.) z Workspace
24	$Table\Phi$	
25	T_s	Periódna vzorkovania

4.4.2 Model DC motora so sériovým buđením

Model DC motora so sériovým buđením vytvorené z rovníc (3.67), (3.68) a (4.1) je takmer zhodný s modelom DC motora s cudzím buđením. Zásadný rozdiel je, že DC motorom so sériovým buđením má spoločný prúd kotvy a budiaceho vinutia, preto do modelu vstupuje menší počet parametrov (tab. 4.4) ako do modelu DC motora s cudzím vinutím.

Závislé veličiny ($\Phi(i_e)$, $L_e(i_e)$) je možné definovať pomocou:

- **Tabuľky** – uloženéj vo Workspace, alebo v dátovom súbore programu Matlab a pri simulácii použitá lineárna interpolácia (4.2)
- **Konštantnej hodnoty** – v oblasti nasýtenia alebo predpokladanej lineárizácie v oblasti pracovného bodu.
- **Nového vstupu** – na ktorý je privedený signál odpovedajúci závislosti danej veličiny.

Takže užívateľské rozhranie je podobné s rozhraním pre DC motor s cudzím buđením (obr. 4.7 a 4.8).

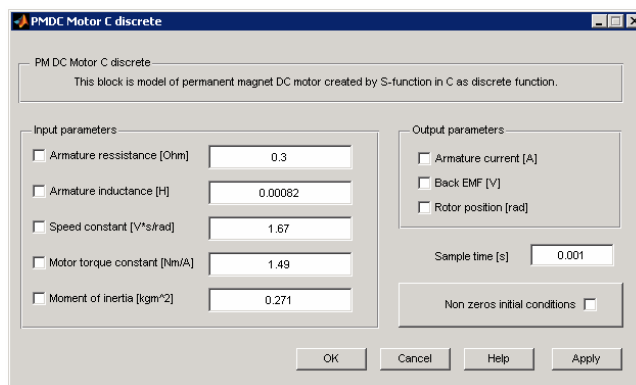
Tabuľka 4.4: Vstupné parametre do modelu DC motora so sériovým buđením

P.č.	Symbol	Popis
1	R_{ae}	Vstupné parametre
2	L_a	
3	L_e	
4	Φ	
5	C	
6	J	
7	$InpR_{ae}$	Definovanie vstupov modelu 0 = parameter 1 = nový vstup $InpL_e$ a $Inp\Phi$: 0 = načítanie z tab. 1 = parameter 2 = nový vstup
8	$InpL_a$	
9	$InpL_e$	
10	$Inp\Phi$	
11	$InpC$	
12	$InpJ$	

P.č.	Symbol	Popis
13	$OutI_a$	Definovanie výstupov modelu (1 / 0)
14	$OutU_i$	
15	v_m	
16	I_a	Počiatočné podmienky
17	ω_m	
18	v_m	
19	$TableL_e$	Názov premennej (tab.) z Workspace
20	$Table\Phi$	
21	T_S	Periódna vzorkovania

4.4.3 Model DC motora s permanentným magnetmi

Jeden z najjednoduchších vytvorených modelov popísaného matematickým vzťahmi (3.65), (3.66) a (4.1). Má tri stavové veličiny a päť základných parametrov (tab. 4.5). Uživatelské rozhranie pre model motora je na obrázku 4.9.



Obrázok 4.9: Uživatelské rozhranie pre model DC s cudzím buđením

Tabuľka 4.5: Vstupné parametre do modelu DC motora s PM

P.č.	Symbol	Popis
1	R_a	Vstupné parametre
2	L_a	
3	K_e	
4	K_t	
5	J	
6	$InpR_a$	Definovanie vstupov modelu 0 = parameter 1 = nový vstup
7	$InpL_a$	
8	$InpK_e$	
9	$InpK_t$	
10	$InpJ$	

P.č.	Symbol	Popis
11	$OutI_a$	Definovanie výstupov modelu (1 / 0)
12	$OutU_i$	
13	ϑ_m	
14	I_a	Počiatočné podmienky
15	ω_m	
16	ϑ_m	
17	T_s	Periódá vzorkovania

4.5 MODEL SPÍNANÉHO RELUKTANČNÉHO MOTORA

Model spínaného reluktančného motora (6/4) bol vytvorený na základe matematických vzťahov (3.74) a (3.78). Pre model je nutné vedieť závislosť magnetického toku na polohe rotora a prúde fáze (obr. 3.16). Predpokladáme, že magnetické toky jednotlivých fáz sa ovplyvňujú veľmi málo a že ich môžeme zanedbať. Preto definovaná závislosť magnetického toku bude platiť pre všetky fáze

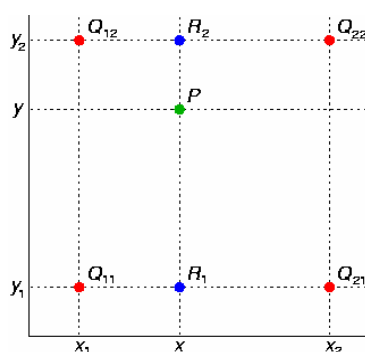
motora rovnako a stačí ju definovať iba jedenkrát. Keďže rotor má 4 póly bude sa závislosť magnetického toku jednej fáze na polohe opakovat' periodicky s periódou $2\pi/4$ tj. 90° . Uvažujme, že závislosť toku na polohe rotora je rovnaká pre pól rotora, ktorý sa približuje, ako pre pól ktorý sa vzdialuje ($f(-10^\circ) = f(+10^\circ)$). Uhol 0° je, keď pól rotora je zosuhlasnený s pólom fáze. Potom stačí definovať závislosť magnetického toku v intervale $\langle 0^\circ; 45^\circ \rangle$ pre polohu a $\langle 0; I_{\max} \rangle$ pre prúd.

Pri tvorbe modelu sme sa uvažovali, že magnetický tok je definovaný tabuľkou pre dve premenne – prúd i_f a poloha θ_f . Ak aktuálne hodnoty polohy a prúdu neodpovedajú hodnotám uvedeným v tabuľke použije sa bilineárna interpolácia [17]. Najprv sa vykoná lineárna interpolácia v smere x (dostávame body R_1 a R_2) a následne v smere y medzi interpolovanými bodmi v smere x , čím dostávame hodnotu v hľadanom bode x, y (θ_f, i_f).

$$f(R_1) = f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_1) \quad (4.3)$$

$$f(R_2) = f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_2) \quad (4.4)$$

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (4.5)$$

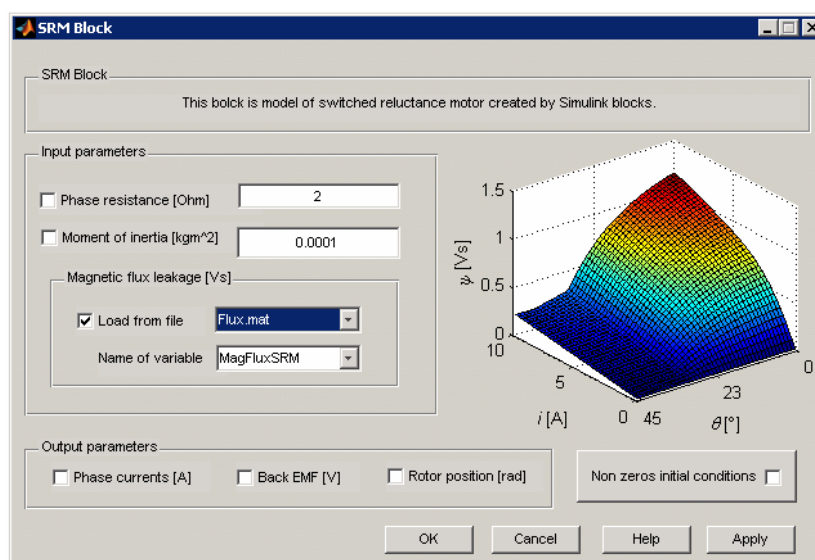


Obrázok 4.10: Bilineárna interpolácia [17]

V matematických vzťahoch pre model SRM sa vyskytujú parciálne derivácie magnetického toku podľa prúdu a polohy. Pre moment motora aj integrál parciálnej derivácie polohy podľa prúdu. Aby v simulácií nemuseli prebiehať zložité operácie pri ich výpočte, z tabuľky $\Psi_f = f(\theta_f, i_f)$ sme vytvorili ďalšie tri tabuľky, kde sa

nachádzajú predpripravené hodnoty parciálnych derivácií a momentu jednej fázy motora.

Keďže samotný motor neobsahuje veľa parametrov, načítanie tabuľky pre $\Psi_f = f(\theta_f, i_f)$ bolo pridané priamo do hlavného GUI. Načítať tabuľku je možné priamo z *Workspace*, alebo z dátového súboru MAT-file, ktorý sa ale musí nachádzať v pracovnom adresári. V pravej časti GUI sa zobrazí trojrozmerný náhľad tejto závislosti (obr. 4.11). Celý postup načítania a správny formát tabuľky je v prílohe B. Po kliknutí na tlačidlo OK, alebo Apply sa predpočítajú tabuľky parciálnych derivácií magnetického toku a momentu jednej fázy motora. Všetky tabuľky sa následne uložia do *Workspace*.



Obrázok 4.11: Užívateľské rozhranie pre model SRM

V tabuľke 4.6 sú parametre, ktoré vstupujú do modelu SRM. Pri zvolenom výstupe prúdov, sa pridá z bloku jeden výstupný port pre všetky tri prúdy statora. Do modelu sa pošlú názvy premenných vo *Workspace*, kde sú uložené predvypočítané tabuľky. Model je doplnený a funkciu kontrolu tabuliek vo *Workspace*, ktorá sa vyvolá pri spustení simulácie (inicializácie). V prípade, že sa vo *Workspace* nenachádzajú hľadané premenných obsahujúce tabelované hodnoty, simulácia sa zastaví a zobrazí sa okno informujúce o chybe. Ak sa tabuľky vytvárali z dátového súboru, inicializačná funkcia sa pokúsi o ich znovu vytvorenie z dátového súboru.

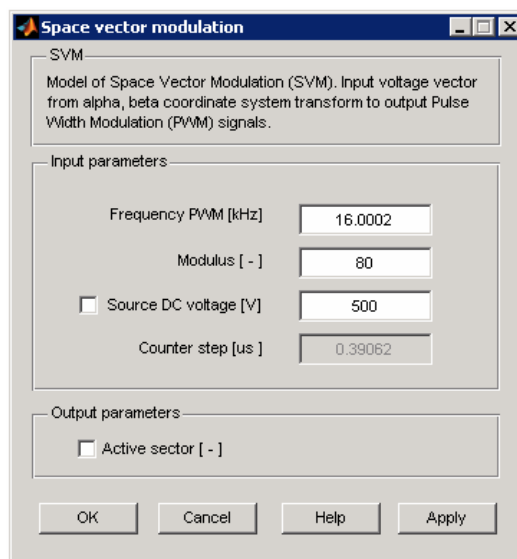
Tabuľka 4.6: Vstupné parametre do modelu SRM

P.č.	Symbol	Popis	P.č.	Symbol	Popis
1	R_a	Vstupné parametre	8	I_a	Počiatočné podmienky
2	J		9	I_b	
3	$InpR_a$	Definovanie vstupov modelu (1 / 0)	10	I_c	
4	$InpJ$		11	ω_m	
5	$Outl_{abc}$	Definovanie výstupov modelu (1 / 0)	12	θ_m	
6	$OutU_i$		13	$Tabled\theta$	Tabuľky s predvypočítanými hodnotami
7	θ_m		14	$Tabledi$	
			15	$TableTe$	
			16	T_s	Periódá vzorkovania

4.6 MODEL SPACE VECTOR MODULATION

Celý algoritmus pozostávajúci z troch krokov, ktorý bol použitý pre vytvorenie modelu SVM je v kapitole 3.6.2. Na rozdiel od ostatných modelov, ktoré boli vytvorené v piatich variantoch, je model SVM vytvorený iba ako diskretný cez S-function v jazyku C.

Vstupom do modelu je požadované napätie v súradniciach $\alpha\beta$. Ďalším voliteľný vstupom je napätie jednosmerného zdroja, ktorý je spínaný PWM. Výstupom sú riadiace signály a,b,c pre PWM. Voliteľným výstupom je, ktorý zo šiestich sektorov je aktívny.



Obrázok 4.12: Užívateľské rozhranie pre model SVM

V GUI sa nastaví frekvencia PWM, *modulus* (1/2 počtu krokov počítadla z ktorých sa skladá jedná perióda PWM). Pre informáciu sa zobrazí ešte doba jedného kroku počítadla.

4.7 ĎALŠIE MODEL Y

Pre simuláciu a podporu modelov bolo potrebné vytvoriť ďalšie bloky. Pre tieto bloky bolo pomocou masky (kap. 2.5) vytvorené užívateľské rozhranie a bloky sa pridali do knižnice.

4.7.1 Transformácia súradníc

Na základe rovníc transformácie súradnicového systému (kap. 3.1) bolo vytvorené šesť blokov pre transformácie súradníc.

- Z abc do $\alpha\beta$ – Forward Clarke Transform (3.2)
- Z $\alpha\beta$ do abc – Reverse Clarke Transform (3.3)
- Z $\alpha\beta$ do dq – Parkova transformácia (3.7)
- Z dq do $\alpha\beta$ – inverzná Parková Transformcia (3.8)
- Z abc do dq – spojenie oboch transformácií (3.9)
- Z dq do abc – spojenie oboch transformácií (3.10)

Bloky transformácie boli vytvorené cez blok, ktorý umožňuje napísať priamo matematický výraz. Navyše boli transformácie vytvorené v dvoch variantoch. S jedným vstupom a výstupom, alebo individuálnym vstupom a výstupom pre každú zložku súradnice.

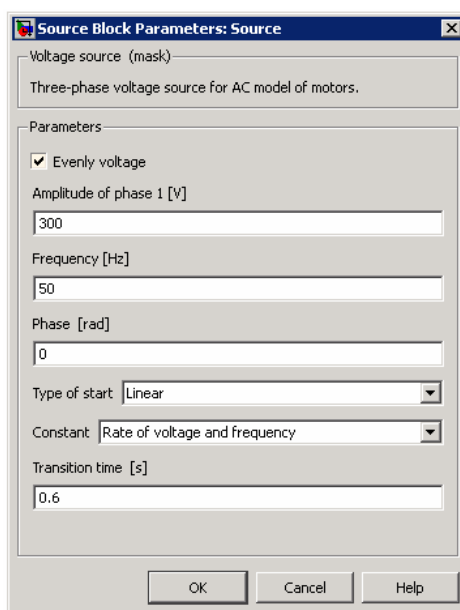
4.7.2 Trojfázový napäťový zdroj

Pre napájanie trojfázových AC motorov bol vytvorený generátor trojfázových signálov podľa rovnice (3.11) cez základné bloky programu Matlab Simulink. Užívateľské rozhranie bolo vytvorené cez masku. Predpokladá sa, že zdroj je symetrický tj. amplitúdy signálov sú rovnaké, takže sa zadaná hodnota amplitúdy použije pre všetky generované signály. Samozrejme bola ponechaná možnosť, nastaviť amplitúdu pre každú fázu samostatne. Ďalšími parametrami sú frekvencia a fázový posun. Na výber sú tri možnosti štartu generátoru:

- Okamžité

- Lineárne narastajúce signály počas určenej doby
- Kvadraticky narastajúce signály počas určenej doby

Pri zvolení lineárne či kvadraticky narastajúcich generovaných signálov je ešte na výber, akým spôsobom to bude prebiehať. Môže byť zvolená konštantná frekvencia, alebo konštantný pomer frekvencie a amplitúdy, čo je vhodné obzvlášť pri napájaní asynchrónneho motora. Užívateľské rozhranie generátoru je na nasledujúcom obrázku.



Obrázok 4.13: Užívateľské rozhranie trojfázového generátoru

4.7.3 Jednoduchý model výkonovej elektroniky – PWM

Pre použitie modelu SVM v simulácií je nutné doplniť k modelu SVM aj blok výkonovej elektroniky spínanej PWM. Model bol realizovaný na základe vzťahu (3.80). Užívateľské rozhranie obsahuje len jeden parameter a tým je hodnota jednosmenného napätie, ktoré spína PWM, prípadne vytvorenie vstupu pre definovanie jednosmenného napájacieho napätia.

4.7.4 Blok pre normalizáciu polohy

Výstup polohy z modelov predstavuje prostú integráciu rýchlosti a výstup sa stále zvyšuje. Bol vytvorený blok, ktorý normalizuje polohu v rad z intervalu $(-\infty, \infty)$ do intervalu $(-\pi, \pi)$.

5. IMPLEMENTÁCIA VYBRANÝCH MODELOV NA PROCESORE

Na procesore rady Freescale 56F800E Hybrid Controller, boli implementované modely: DC motora s permanentnými magnetmi a model synchronného motora s permanentnými magnetmi.

5.1 PREPOJENIE MODELU NA PROCESORE S PROSTREDÍM MATLAB SIMULINK

Vytvorenie modelu bez možnosti simulácie, by nemalo veľký prínos. Pred samotnou tvorbou modelu na procesore musíme najprv zaistiť spôsob overenia funkčnosti modelu a možnosti simulácie. Najvýhodnejšou možnosťou je prepojenie modelu na procesore od firmy Freescale s prostredím Matlab Simulink cez toolbox SFIO¹.

Toolbox SFIO umožňuje nie len pridať blok modelu vytvoreného na procesore a zabezpečiť s ním komunikáciu, ale aj z časti generovať kód pre procesor, aby boli správne spárované vstupy a výstupy modelu s prostredím Matlab Simulink. V bloku modelu SFIO sa nastaví názov funkcie (knihovni), kde bude vytvorený model, rozmery vstupov a výstupov a definovanie dátových typov. Pre oba modely boli použité vstupy a výstupy s dátovým typom *short* (16b). Vstupné signály je nutné správne normalizovať, aby vstupujúca hodnota do modelu nepresiahla hodnotu z intervalu $\langle -2^{15}; 2^{15} - 1 \rangle$. Súčasne vstupujúci signál by sa mal pohybovať v celom intervale, aby bola zabezpečená čo najväčšia možná rozlíšiteľnosť signálu. Obdobne aj výstupné signály z modelu je nutné denormalizovať z intervalu $\langle -2^{15}; 2^{15} - 1 \rangle$ do intervalu odpovedajúcim skutočným hodnotám veličín. Pre vstup sa normalizácia určí podľa vzťahu

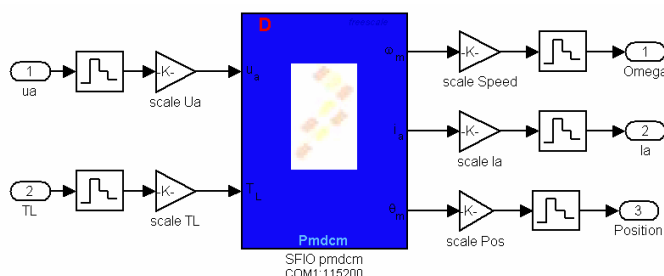
¹ toolbox SFIO bol vyvinutý na UAMT FEKT v Brne pre prepojenie procesoru rady Freescale a programu Matlab Simulink

$$x_{norm} = \frac{2^{15}}{x_{max}} \quad (5.1)$$

a pre výstupný signál

$$y_{denorm} = \frac{y_{max}}{2^{15}} \quad (5.2)$$

Implementovaný model je diskretný a preto sú vstupy a výstupy doplnené o pridržiavače a tvarovače s periódou vzorkovania, ako je naznačené na nasledujúcom obrázku.



Obrázok 5.1: Blok reprezentujúci model na procesore pripojený cez SFIO

SFIO toolbox obsahuje blok *Breakpoint*. Vložení makra *SFIO_Breakpoint()* do zdrojového kódu a využitím tohto bloku, sa pozastaví simulácia pri aktivácii tohto bloku a ako aj vykonávaný kód procesoru. Aktivácia bloku nastane asynchrónne (dvojklikom na blok), alebo synchrónne (určitou udalosťou). To dáva značne možnosti pri ladení programu.

5.2 MODEL DC MOTORA S PM

5.2.1 Implementácia modelu na procesore

Ako bolo uvedené v predchádzajúcej kapitole, cez SFIO toolbox sa dá vygenerovať kód so štruktúrou vstupov, výstupov a predefinovanými funkciami, takže stačí kód doplniť o matematický popis motora, stavy a premenné. Pre vytvorenie modelu sme však vytvorili samostatnú funkciu a štruktúry (vstupov, výstupov, stavov), ktoré sa nemusia zhodovať s štruktúrou bloku definovaného v programe Matlab Simulink. Tým je možné rozšíriť blok a program na procesore o ďalšie vstupy, výstupy a stavy ako má samotný model napr. o meranie doby vykonávania funkcie a pod.

Vzťahy pre matematický popis DC motora s permanentnými magnetmi sa musia upraviť pre implementáciu na procesore do diferenčných rovníc a normalizovať podľa maximálnych hodnôt veličín. Po úprave rovníc (3.65), (3.66) a (4.1) dostávame vzťahy

$$i_a(k+1) = i_a(k) + K_1 u_a(k) - K_2 i_a(k) - K_3 \omega_m(k), \quad (5.3)$$

$$\omega_m(k+1) = \omega_m(k) + K_4 i_a(k) - K_5 T_L(k), \quad (5.4)$$

$$\vartheta_m(k+1) = \vartheta_m(k) + K_6 \omega_m(k), \quad (5.5)$$

kde

$$K_1 = \frac{T_S u_{a \max}}{L_a i_{a \max}}, \quad K_2 = \frac{T_S R_a}{L_a}, \quad K_3 = \frac{T_S K_e \omega_{m \max}}{L_a i_{a \max}}, \quad K_4 = \frac{T_S K_t i_{a \max}}{J \omega_{m \max}},$$

$$K_5 = \frac{T_S T_{L \max}}{J \omega_{m \max}}, \quad K_6 = \frac{T_S \omega_{m \max}}{\vartheta_{m \max}} \quad (5.6)$$

Veličiny s indexmi *max* sú maximálne hodnoty veličín, ktoré sú z dôvodu normalizácie vstupu, či výstupu a T_S je perióda vzorkovania. Základné matematické operácie sú realizované pomocou *intrinsic function* [3]. Vstupné a výstupné veličiny sú definované ako 16b (Frac16). Výsledkom pri násobení dvoch čísel (Frac16) je 32b číslo (Frac32). Stavové veličiny sú preto definované ako 32b (Frac32). Keďže zmenšíme zaokrúhľovanie stavových veličín, zvýši sa tým presnosť modelu

Hodnoty konštánt sa však v závislosti na parametroch môžu pohybovať v úplne inom rozmedzí ako v intervale $\langle -1; 1 \rangle$. Ak hodnota konštanty bude veľmi malá – blízka nule, tak sa neprejaví vôbec jej vplyv. Ak bude hodnota veľká, bude hodnota obmedzená. Preto sa hodnota *bitovo posunie* o toľko miest (*shiftK_i*), aby bola konštanta maximálna možná z intervalu $\langle -1; 1 \rangle$.

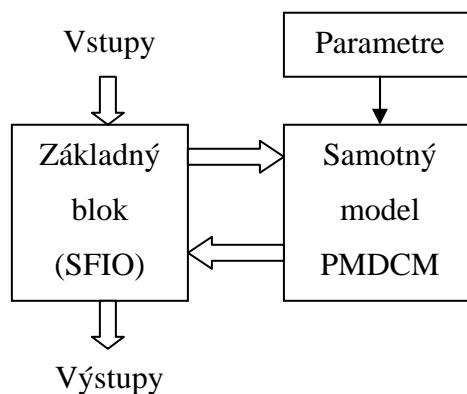
Výsledok súčinu „posunutej“ konštanty so vstupnou, či stavovou veličinou je nutné spätne *bitovo posunúť*. Preto sú súčty/rozdiely dielčích výrazov zoradené podľa počtu bitov, o ktoré sú posunuté konštanty dielčích výrazov. Dielčí výsledok sa bitovo posunie o rozdiel medzi zoradenými bitovými posunmi, aby sa takto pripravila per nasledujúcu operáciu. Princíp *bitového posunu* (\ll) prúdu kotvy (5.3) je znázornený na nasledujúcich rovniciach pre situáciu $\text{shift}K_1 < \text{shift}K_2 < \text{shift}K_3$

$$i_{a3} = (-K_3 \omega_m) \ll (\text{shift}K_3 - \text{shift}K_2) \quad (5.7)$$

$$i_{a2} = (i_{a3} - K_2 i_a) \ll (\text{shift}K_2 - \text{shift}K_1) \quad (5.8)$$

$$i_a = (i_{a2} - K_1 u_a) \ll \text{shift}K_1 \quad (5.9)$$

Pre vytvorený model sa vytvorili 3 knihovne. V prvej je knihovne je vytvorená pre blok v programe Matlab Simulink (SFIO). Druhá knihovňa obsahuje samotný model a definovanie štruktúry dát pre model. V tretej knihovne sa nachádzajú parametre pre model s definovanými bitovými posunmi konštánt. V modeli sa konštanty vypočítajú iba jedenkrát a to pri inicializácii stavov a uložia sa do príslušnej štruktúry.

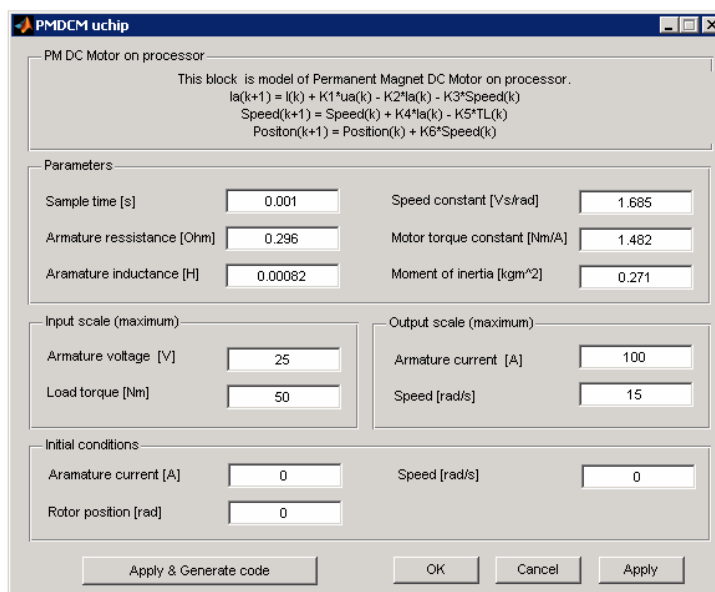


Obrázok 5.2: Implementácia modelu na procesore

5.2.2 Uživatelské rozhranie a generovanie kódu pre model

Blok modelu DC motora s permanentnými magnetmi je tvorený blokmi ako je na obrázku 5.1. Pre tento blok bolo vytvorené GUI, ktoré nastavuje pre tvarovače a pridržiavače taktovaciu periódu, taktiež hodnoty pre bloky normalizácie a denormalizácie. Okrem toho generuje kód pre knihovnu parametrov. Zo zadaných parametrov v rozhraní sa predpočítajú konštanty, z ktorých sa určí bitový posun konštánt a poradie, v akom sa majú počítať dielčie výrazy jednotlivých diferenčných rovníc. Ukážka navrhnutého užívateľského rozhrania je na obrázku 5.3.

Rozhranie sa skladá z troch častí. Z parametroch motora, parametrov pre normalizáciu vstupov a denormalizáciu výstupov. Tlačidlo Apply & Generate code uloží vykonané zmeny a vygeneruje kód, ktorý sa vypíše v okne programu Matlab. Tento kód stačí skopírovať a nahradiť predchádzajúci kód definujúci parametre modelu. Tlačidlá OK, Apply a Cancel fungujú ako je to uvedené v kap.4.1.2.



Obrázok 5.3: GUI pre blok DC motora s PM realizovaného na procesore

5.3 MODEL SYNCHRÓNNEHO MOTORA S PM

Tvorba modelu na procesore je obdobná predchádzajúcemu modelu. Matematicky popis motora je určený rovnicami (3.58) až (3.61) a (3.26), ktoré je nutné upraviť do diferenčných rovníc a súčasne zahrnúť normalizačné konštanty. Po úpravách dostávame vzťahy

$$i_d(k+1) = i_d(k) + K_1 u_d(k) - K_2 i_d(k) + K_3 \omega_e(k) i_q(k), \quad (5.10)$$

$$i_q(k+1) = i_q(k) + K_4 u_q(k) - K_5 i_q(k) - K_6 \omega_e(k) i_d(k) - K_7 \omega_e(k), \quad (5.11)$$

$$\omega_e(k+1) = \omega_e(k) + K_8 i_q(k) - K_9 i_d(k) i_q(k) - K_{10} T_L(k), \quad (5.12)$$

$$\vartheta_e(k+1) = \vartheta_e(k) + K_{11} \omega_e(k), \quad (5.13)$$

kde

$$\begin{aligned} K_1 &= \frac{T_S u_{\max}}{L_d i_{\max}}, \quad K_2 = \frac{T_S R_S}{L_d}, \quad K_3 = \frac{T_S L_q \omega_{e \max}}{L_d}, \quad K_4 = \frac{T_S u_{\max}}{L_q i_{\max}}, \\ K_5 &= \frac{T_S R_S}{L_q}, \quad K_6 = \frac{T_S L_d \omega_{e \max}}{L_q}, \quad K_7 = \frac{T_S \psi_f u_{\max}}{L_q i_{\max}}, \quad K_8 = \frac{3 T_S P_P^2 \psi_f i_{\max}}{2 J \omega_{e \max}}, \\ K_9 &= \frac{3 T_S P_P^2 (L_d - L_q) i_{\max}^2}{2 J \omega_{e \max}}, \quad K_{10} = \frac{T_S P_P T_{L \max}}{J \omega_{e \max}}, \quad K_{11} = \frac{T_S \omega_{e \max}}{\vartheta_{e \max}}. \end{aligned} \quad (5.14)$$

Rovnice pre model sú v dq súradniciach, ale vstupy a výstupy modelu nie sú v dq súradniciach. Do modelu sa musia pridať transformácie súradníc. Knihovňa *MCLIB* (MCLIB_56800ESDM.lib) podporuje prácu s *fractional arithmetic*, transformácie súradníc a goniometrické funkcie. Pridaním knihovni do modelu, sa transformácie súradníc jednoducho implementujú. Výstupy z modelu sú prúdy statora, uhlová rýchlosť rotora a poloha rotora. Na rozdiel od stavových veličín elektrickej uhlovej rýchlosti a polohy, preto sa musia stavy pri priradení výstupom prepočítať.

Proces normalizácie a denormalizácie je rovnaký, ako u modelu DC motora s PM a aj rozhranie funguje podobne. Po zadaní parametroch do GUI sa vygeneruje kód pre knihovňu parametroch, ktorý nahradí predchádzajúce nastavenie parametrov.

PMSM uchip

PMSM on processor

This block is model of Permanent Magnet Synchronous Motor proceed on processor.

$$\begin{aligned} Id(k+1) &= Id(k) + K1*ud(k) - K2*Id(k) + K3*Speed(k)*Iq(k) \\ Iq(k+1) &= Iq(k) + K4*uq(k) - K5*Iq(k) - K6*Speed(k)*Id(k) - K7*Speed(k) \\ Speed(k+1) &= Speed(k) + K8*Iq(k) - K9*Id(k)*Iq(k) - K10*TL(k) \\ Positon(k+1) &= Positon(k) + K11*Speed(k) \end{aligned}$$

Parameters

Sample time [s]	0.0002	Inductance in d axis [H]	0.0009
Stator ressistance [Ohm]	0.273	Inductance in q axis [H]	0.0005
Mutual flux [Vs]	0.00867	Moment of inertia [kgm ²]	3e-006
Rotor pole pairs [-]	3		

Input scale (maximum)

Input voltage [V]	8
Load torque [Nm]	0.2

Output scale (maximum)

Current [A]	15
Speed [rad/s]	150

Initial conditions

Current in d axis [A]	0	Speed [rad/s]	0
Ccurrent in q axis [A]	0	Rotor position [rad]	0

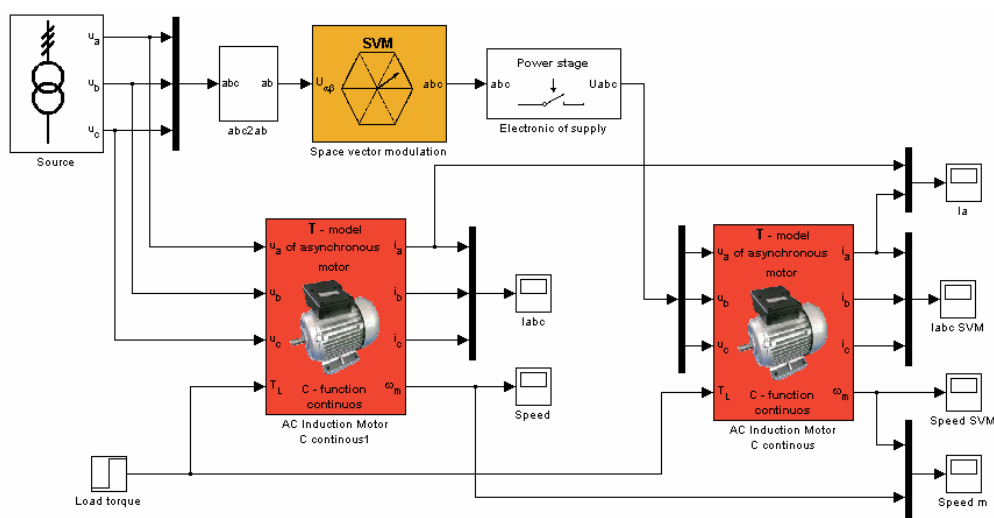
Buttons: Apply & Generate code, OK, Cancel, Apply

Obrázok 5.4: GUI pre blok PMSM realizovaného na procesore

6. SIMULÁCIA MODELOV

6.1 SIMULÁCIA MODELOV SVM A ASYNHRÓNNEHO MOTORA

Na obrázku 6.1 sa nachádza simulačná schéma použitá pre zistenie odoziev výstupov modelu. Pre simuláciu boli použité dva modely ASM s rovnakými parametrami uvedené v tabuľke 6.1. Jeden model ASM je napájaný z trojfázového zdroja napätia. Druhý model je napájaný DC zdrojom s PWM, riadený SVM. SVM by mala generovať riadiace signály pre PWM, aby výstup zdroja odpovedal trojfázovému zdroju, ktorý napája prvý model ASM. Blok trojfázového zdroja má nastavenú amplitúdu 250V, kmitočet 50Hz. Napätie lineárne narastá z nuly na menovité hodnoty počas doby 0,5s so zachovaním konštantného pomeru kmitočtu a amplitúdy (obr. 6.2). Blok SVM bol nastavený pre frekvenciu PWM 1kHz, *modulus* 80 a napätím DC zdroja 500V.



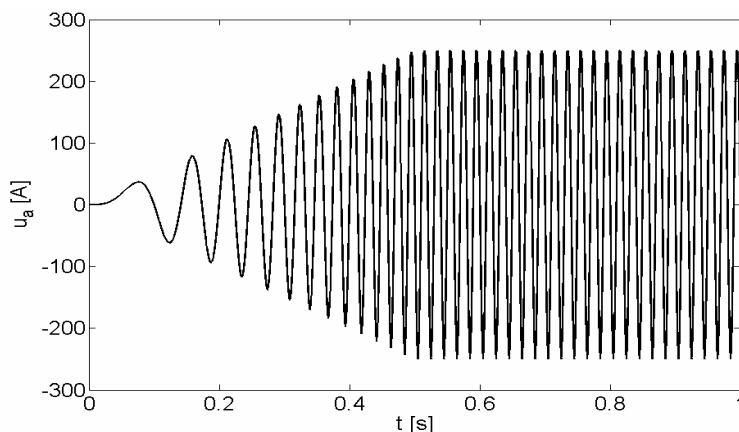
Obrázok 6.1: Simulačná schéma pre model ASM a zdroja riadeného SVM

Tabuľka 6.1: Parametre ASM použitého pri simulácii

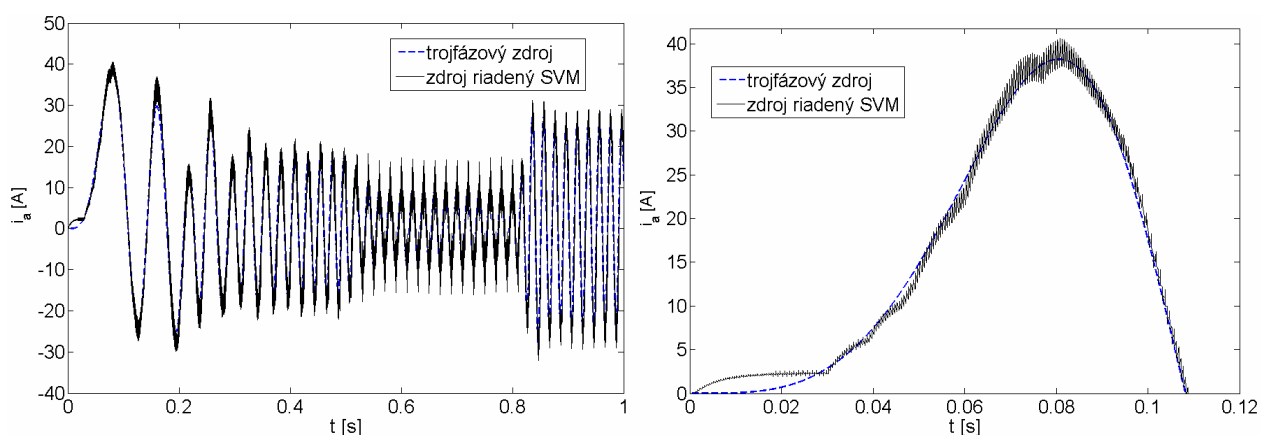
Parametre T-modelu	$L_{S\sigma}$ [mH]	$L_{R\sigma}$ [mH]	L_m [mH]	R_S [Ω]	R_R [Ω]	J [kgm ²]	P_P [-]
	2,5	2,5	84,7	0,531	0,408	0,1	2

Pre porovnanie vlastností asynchrónneho motora napájaného z DC zdroja riadeným SVM a „klasickým“ zdrojom, sú oba priebehy výstupných signálov

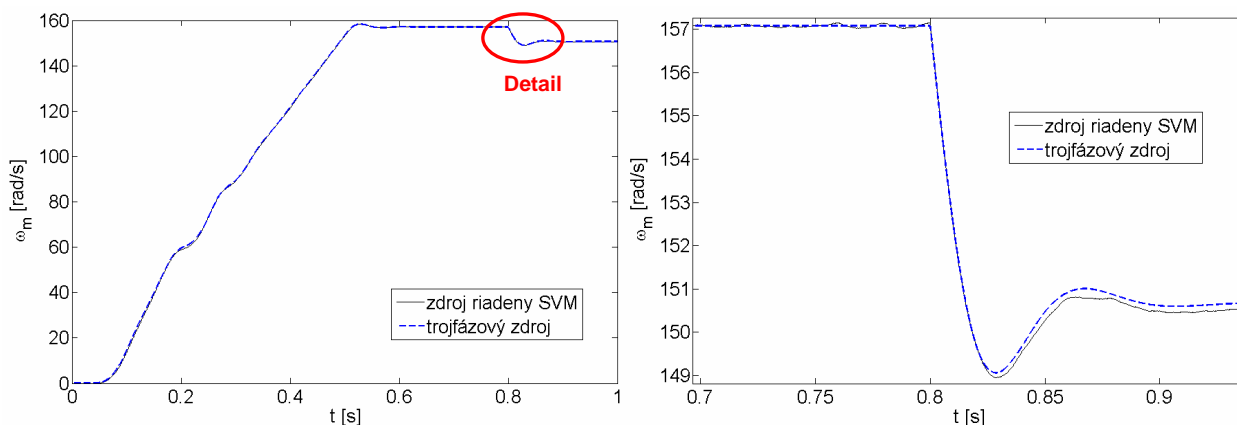
modelov asynchronných motorov v spoločných grafických signálov, ktoré sa nachádzajú na obrázkoch 6.3 a 6.4. V čase 0,8s sa zmení moment záťaže skokom z hodnoty 0 na hodnotu 50Nm.



Obrázok 6.2: Napätie fáze a generovaného zdrojom trojfázového napätia



Obrázok 6.3: Prúd satorového vinutia a ASM, vpravo detail začiatku

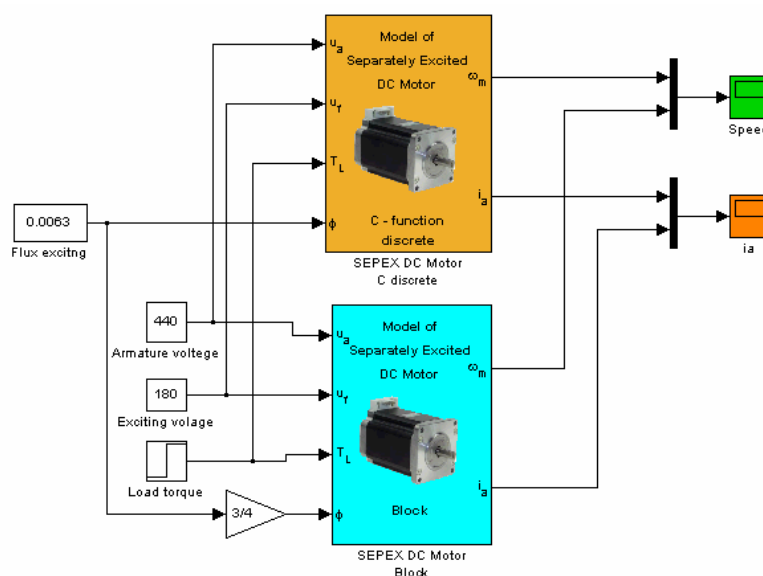


Obrázok 6.4: Uhlová rýchlosť rotora ASM, vpravo detail

Z priebehov je vidieť, že model ASM napájaného DC zdrojom riadený metódou SVM, má odozvy podobné ako model napájaný trojfázovým zdrojom. Pri zaťažení modelu motora momentom 50Nm, klesli otáčky (ω_s) z 157 rad/s na 151rad/s.

6.2 SIMULÁCIA MODELU DC MOTORA S CUDZÍM BUDENÍM

V simulácii DC motora s cudzím buđením sme použili dva bloky modelov. Diskrétny variant (C-function discrete) a spojitý variant, realizovaný pomocou základných blokov (obr. 6.5). Pre magnetický tok bol vytvorený nový vstup a jeho hodnota bola definovaná konštantou (tab. 6.2). Do bloku spojitý varianty (SEPEX DC Motor Block) vstupuje $\frac{3}{4}$ definovanej hodnoty magnetického toku. V čase 0,3s sa zmení moment záťaže skokom z hodnoty 0 na hodnotu 100Nm. Odozvy modelov sú na obrázku 6.6.

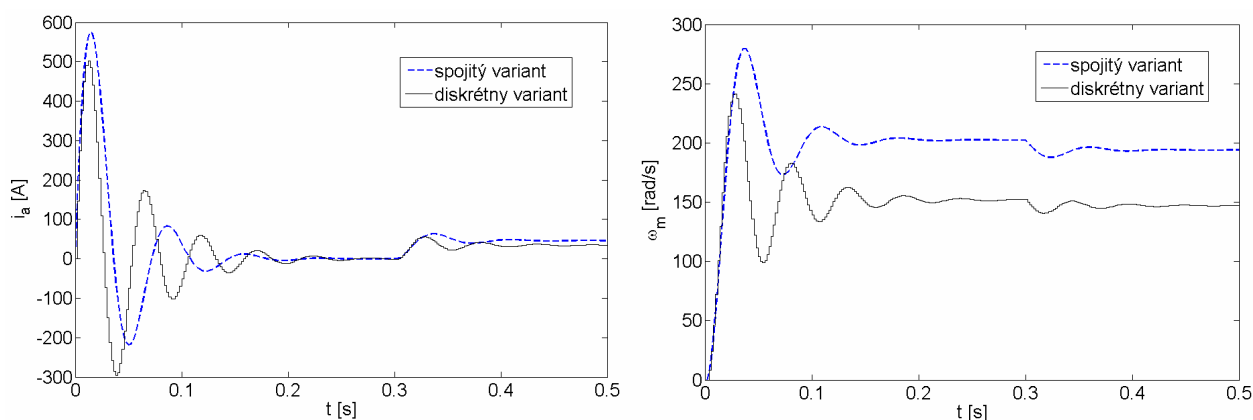


Obrázok 6.5: Simulačná schéma pre model DC motora s cudzím buđením

Tabuľka 6.2: Parametre DC motorov so cudzím vinutím použitých pri simulácii

Parameter	U_a [V]	U_b [V]	R_a [Ω]	R_b [Ω]	L_a [mH]	L_b [H]	Φ [mWb]	J [kgm ²]
Spojité variant	440	180	0,4	60	6	20	6,3	0,1
Diskrétny variant	440	180	0,4	60	6	20	$\frac{3}{4} \cdot 6,3$	0,1

Pozn. $C = 460$, $T_S = 2\text{ms}$



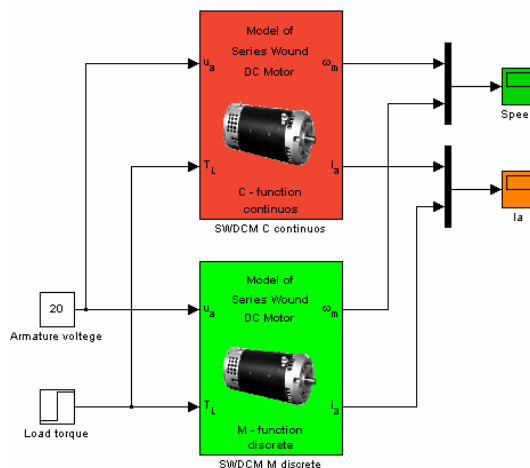
Obrázok 6.6: Vľavo priebeh prúdov, vpravo odozva rýchlosti SEPEX DC motorov

Pri odbudzovaní motora ($\frac{3}{4}$ menovitej hodnoty budiaceho toku – spojitý variant) má model motora vyššiu rýchlosť (obr. 6.6), čo odpovedá očakávanej odozve. Zaťažením motora momentom 100Nm klesli otáčky rotora a zväčšil sa prúd.

6.3 SIMULÁCIA MODELU DC MOTORA SO SÉRIOVÝM BUDENÍM

Pre simuláciu sme použili dva modely. C-function continuous a M-function discrete s periódou vzorkovania 20ms a konštantou motora $C = 0,15$. Simulačná schéma je na nasledujúcom obrázku.

Oba modely majú rovnaké parametre a líšia sa len implementáciou matematického popisu. Použité parametre sú v tab. 6.3. Moment záťaže sa v čase 3s zmení skokom z hodnoty 0Nm na 0,1Nm., a odozvy výstupov na obr. 6.8.

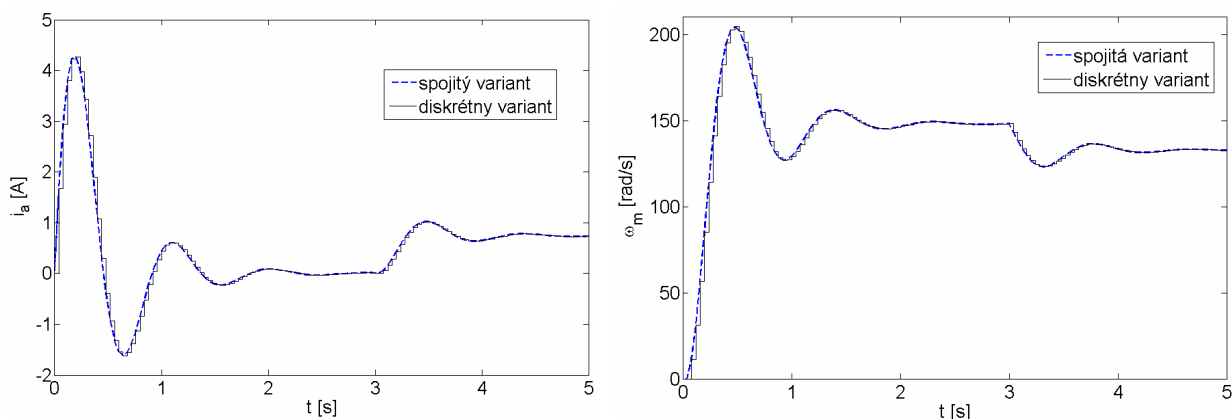


Obrázok 6.7: Simulačná schéma pre model DC motora so sériovým budením

Tabuľka 6.3: Parametre DC motorov so sériovým vinutím použitých pri simulácii

Parametre modelov	U_a [V]	R_a [Ω]	L_a [mH]	L_b [mH]	Φ [Wb]	J [kgm ²]
	20	2,8	80	400	0,9	0,1

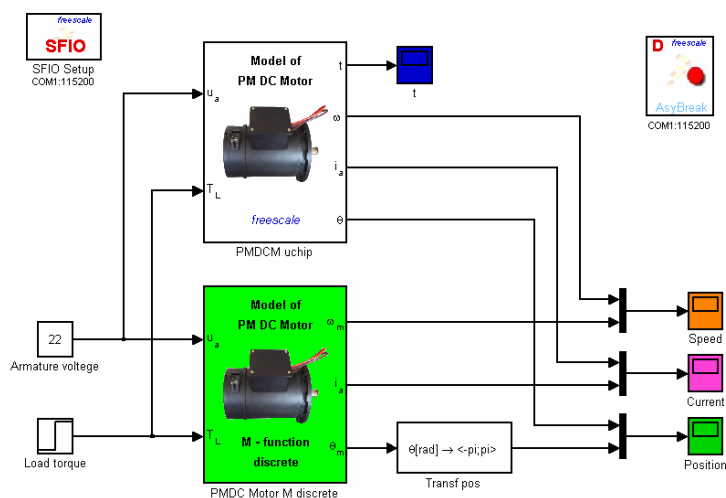
Pozn. $C = 460$, $T_S = 2\text{ms}$



Obrázok 6.8: Vľavo priebeh prúdov, vpravo odozva rýchlosti SWDCM modelu

6.4 SIMULÁCIA MODELU DC MOTORA S PERMANENTÝMI MAGNETMI

Model DC motora s permanentným magnetmi bol implementovaný nie len priamo v programe Matlab Simulink, ale aj na procesore. V simulácii sa nachádzajú oba modely (obr. 6.9) a môžeme porovnať odozvy výstupov oboch rôznych implementácií využitím toolboxu SFIO.

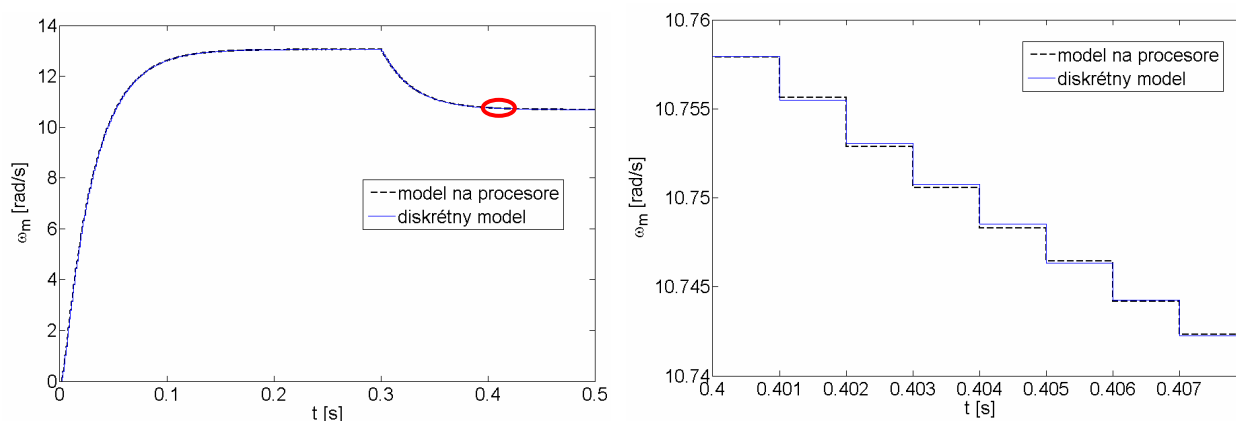


Obrázok 6.9: Simulačná schéma pre model DC motora s PM

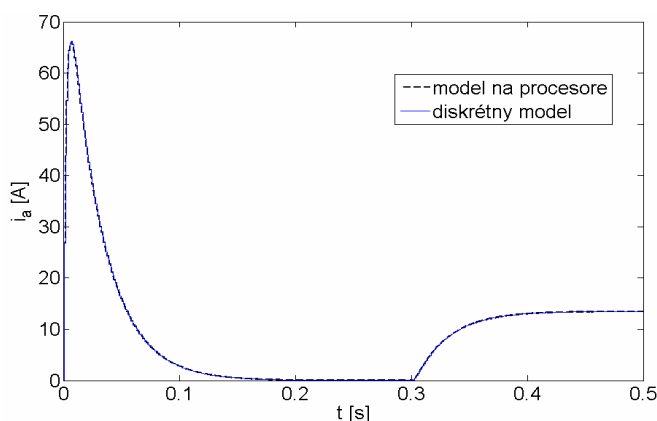
K modelu implementovaného na procesore bol pridaný výstup, pre meranie počtu časových impulzov – ako dlho trvá výpočet jedného simulačného kroku modelu na procesore. K porovnaní presnosti, bol druhý model zvolený ako diskretný (C-function discrete), s rovnakou periódou vzorkovania 1ms. Parametre modelov sú zhodne a ich hodnoty sú v tabuľke 6.4. Do modelov vstupuje napájacie napätie a moment záťaže so skokovou zmenou v čase 0,3s z 0Nm na 20Nm. Konštanty pre normalizáciu vstupov a výstupov boli určené na základe ustálených stavov a vstupných signálov ako $u_{amax} = 25V$, $T_{Lmax} = 50Nm$, $i_{amax} = 100A$, $\omega_{max} = 15rad/s$. Odozvy výstupov sú na obrázkoch 6.10 a 6.11

Tabuľka 6.4: Parametre DC motorov so sériovým vinutím použitých pri simulácii

Parametre modelov	U_a [V]	R_a [Ω]	L_a [mH]	K_e [Vs/rad]	K_t [Nm/A]	J [kgm ²]
	22	0,296	8,2	1,685	1,482	0,271



Obrázok 6.10: Pribeh uhlovej rýchlosti modelov PMDC motorov, vpravo detail



Obrázok 6.11: Pribeh prúdov kotvy modelov PMDC motorov

Porovnaním odoziev pre oba druhy implementácie modelu je zrejmé, že výsledné odozvy sú takmer identické, a že proces normalizácie a bitového posúvania konštánt bol implementovaný správne. Doba výpočtu každého simulačného krok na procesoru bola nemenná počas celej simulácie a to 341 časových impulzov hodín procesoru.

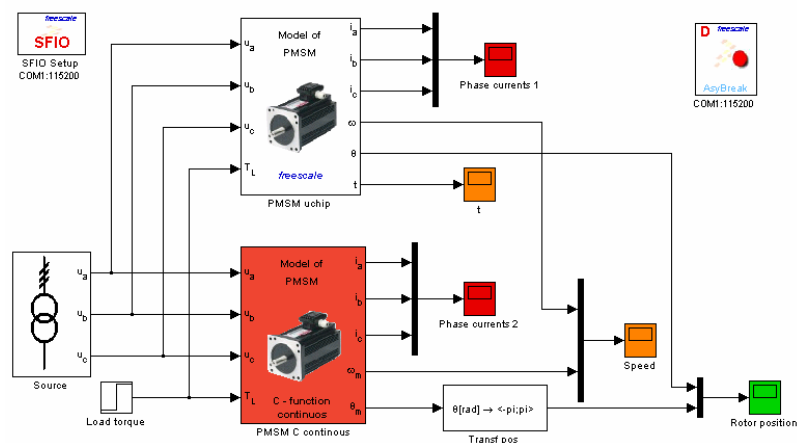
6.5 SIMULÁCIA MODELU PMSM

Simulácia obsahovala dva modely PMSM (obr. 6.12). Prvým modelom bol model PMSM implementovaný na procesore, s pridaným výstupom pre zistenie doby výpočtu jedného simulačného kroku na procesore. Druhým modelom je model implementovaný ako C-function continuous.

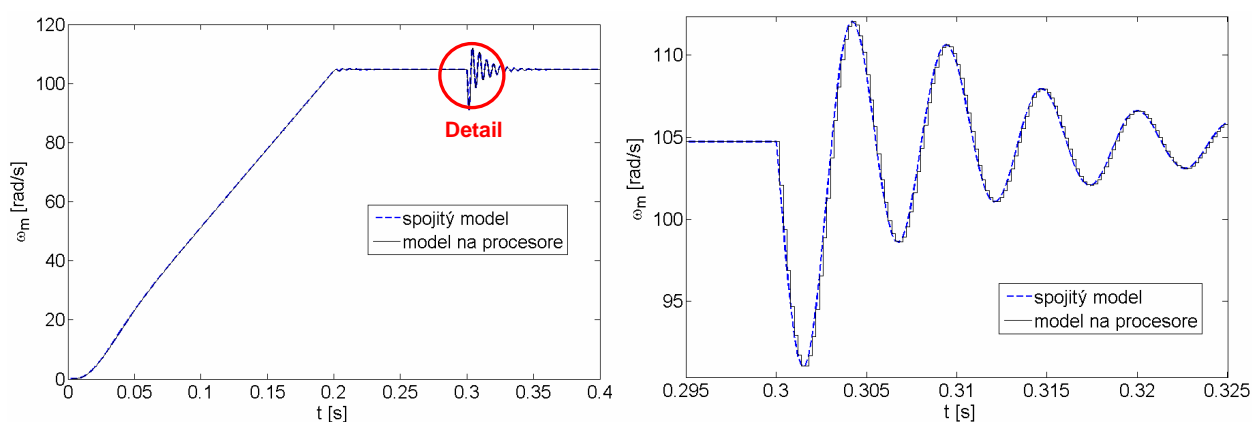
Parametre oboch modelov sú rovnaké a sú uvedené v tabuľke 6.5. Blok trojfázového zdroja, ktorý napája oba modely, má nastavenú amplitúdu 5V, kmitočet 50Hz a fázový posun $\pi/2$. Napätie lineárne narastá z nuly na menovité hodnoty počas doby 0,2s so zachovaním konštantného pomeru kmitočtu a amplitúdy. Konštanty pre normalizáciu vstupov a výstupov boli na základe ustálených stavov a vstupných signálov použité $u_{max} = 8V$, $T_{Lmax} = 0,2Nm$, $i_{max} = 15A$, $\omega_{max} = 150rad/s$.

Tabuľka 6.5: Parametre DC motorov so sériovým vinutím použitých pri simulácii

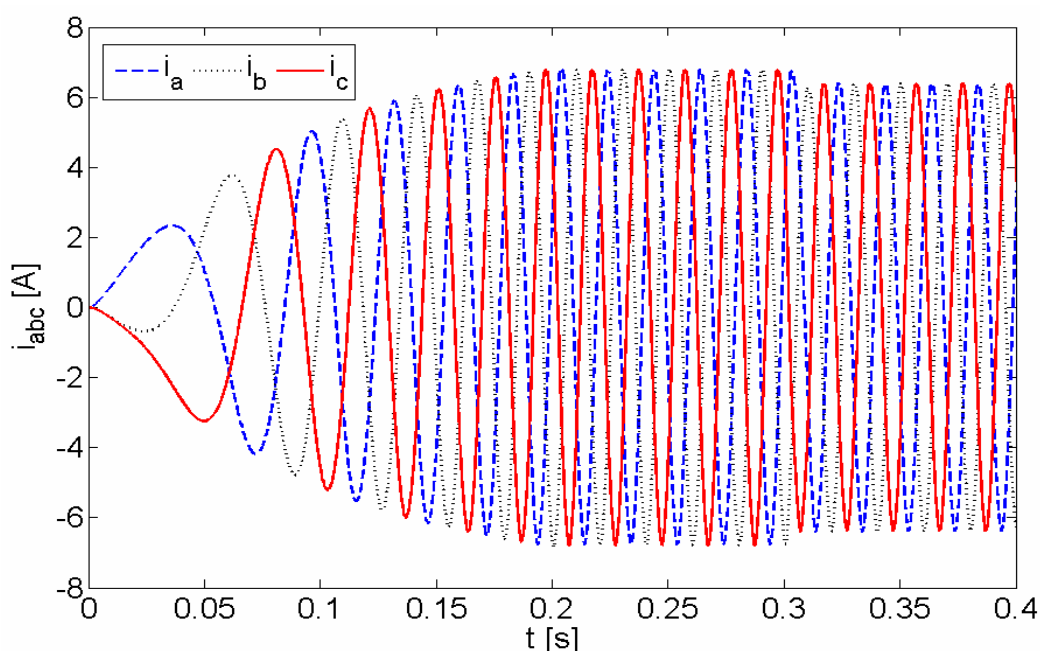
Parametre modelov	$R_s [\Omega]$	$L_d [mH]$	$L_q [mH]$	$\Psi_f [mWb]$	$P_p [-]$	$J [kgm^2]$
	0,273	0,9	0,5	8,67	3	$3 \cdot 10^{-6}$



Obrázok 6.12: Simulačná schéma pre model PMSM



Obrázok 6.13: Uhlová rýchlosť modelov PMSM, vpravo detail



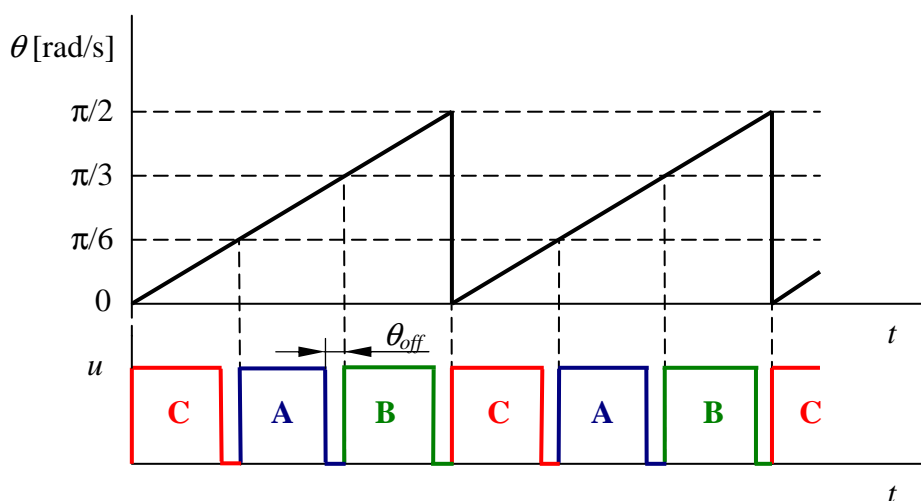
Obrázok 6.14: Priebeh prúdov statora modelu PMSM realizovaného na procesore

Do oboch modelov vstupuje moment záťaže, ktorý v čase 0,3s skokom zmení svoju hodnotu z 0Nm na 0,04Nm. Po odoznení prechodového deja zmeny záťaže sa otáčky ustália na pôvodnej synchronnej rýchlosti (3.28).

Model PMSM realizovaný na procesore kopíruje priebeh modelu vytvoreného ako C-function continuous (obr. 6.13), takže proces normalizácie a bitového posúvania konštánt bol implementovaný správne. Doba výpočtu každého simuláčného kroku na procesore bola 983 časových impulzov hodín a nemenila sa počas celej doby simulácie.

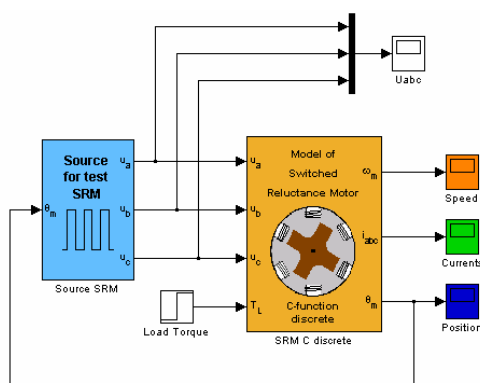
6.6 SIMULÁCIA MODELU SRM

V simulácii spínaného reluktančného motora, bolo nutné vytvoriť zdroj, ktorý by bol vhodný pre tento typ motora. Zdroj nie len že by mal správne spínať napätia pre jednotlivé fázy, ale mal by vypínať napätie fáze tak, aby prúd vinutia stihol klesnúť na nulu skôr, než sa zopne napätie na ďalšej fáze motora. Doba nenulového prúdu určuje aj otáčky motora (3.77).



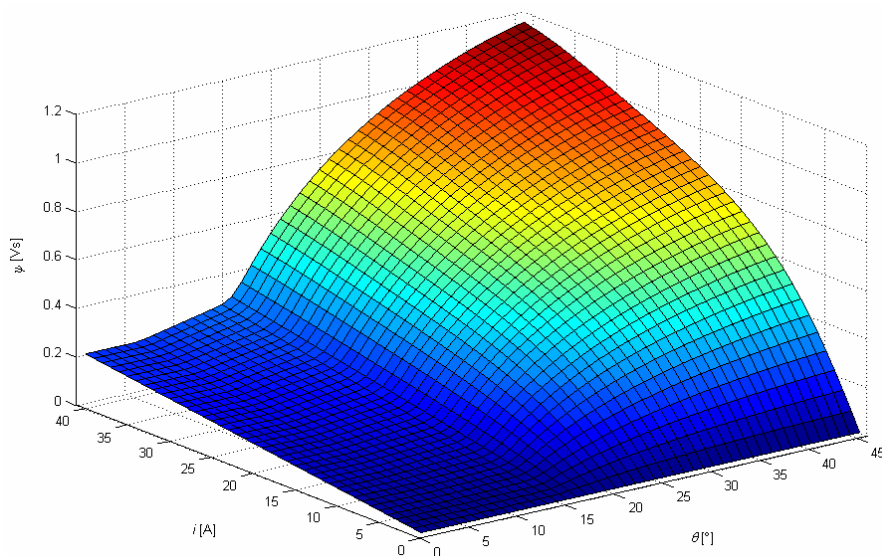
Obrázok 6.15: Priebeh prúdov statora modelu PMSM realizovaného na procesore

Pre zjednodušenie týchto nárokov na spínaný zdroj, sme zaviedli pevný uhol θ_{off} , ktorý určuje dobu aktívneho napätia pre jednotlivé vinutia. Poloha rotora sa transformuje do intervalu $<0; \pi/2$), keďže rotor má 4 póly. Spínanie napätia zdroja ja odvodené na základe poradia spínania vinutí (obr. 3.15) a aktuálnej polohy ako je znázornené na obrázku 6.15. Hodnota amplitúdy DC zdroja u bola zvolená 20V a uhol $\theta_{off} = \pi/16$.



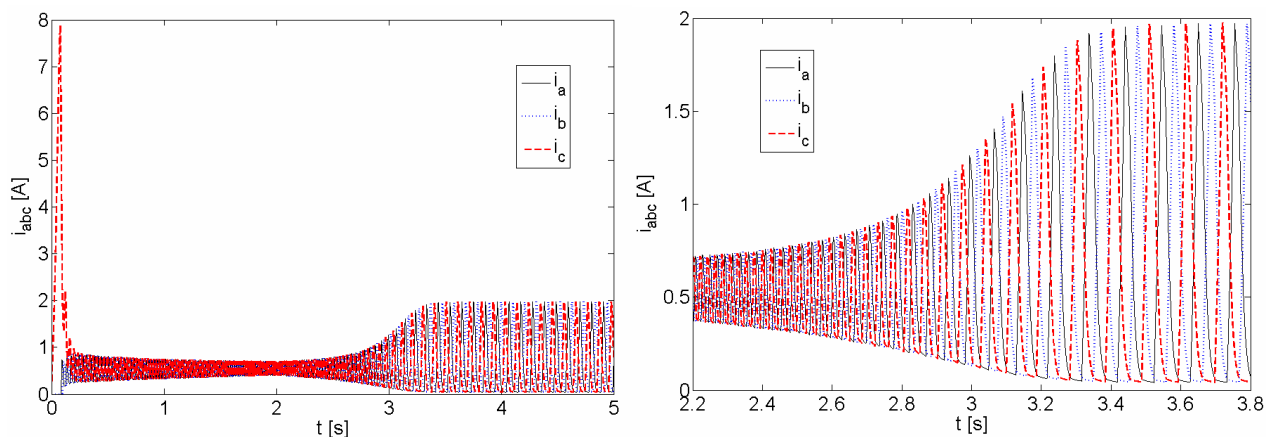
Obrázok 6.16: Simulačná schéma pre model PMSM

Základným parametrom spínaného reluktančného motora je magnetizačná charakteristika, ktorá definuje celkové vlastnosti motora. Magnetický tok je závislý na polohe a prúde. Tieto dáta sme však nemali k dispozícii. Preto sme pre simuláciu vytvorili dáta na základe obrázku 3.16. Grafické znázornenie vytvorených dát je na nasledujúcom obrázku.



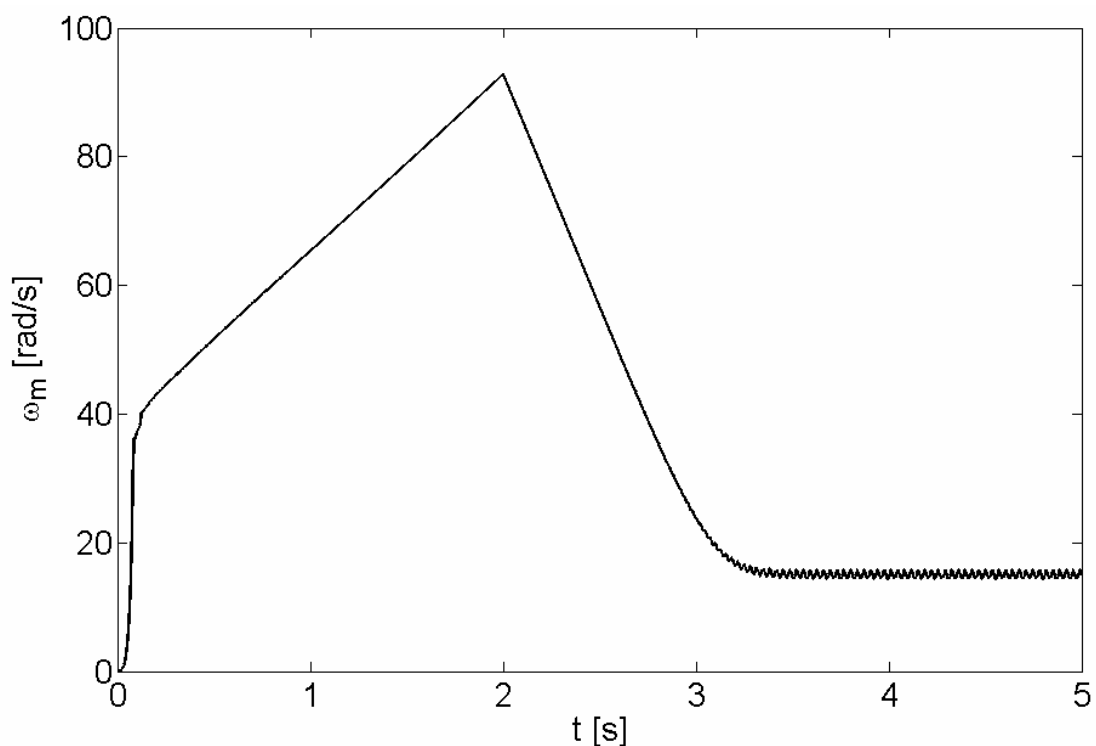
Obrázok 6.17: Magnetizačná charakteristika SRM použitá pre simuláciu

Simulačné schéma použitá pre simuláciu modelu SRM je na obrázku 6.16. Ďalšími parametrami modelu okrem použitých vytvorených dát, boli $R_f = 2\Omega$ a moment zotrvačnosti $J = 1 \cdot 10^{-4} \text{ kgm}^2$. Do modelu vstupuje moment záťaže. V čase 2s sa jeho hodnota zmení skokom z 0Nm na 0,01Nm.



Obrázok 6.18: Priebeh prúdov statora modelu SRM, vpravo detail

Odozvy výstupov modelu zo simulácie sú na obrázkoch 6.18 a 6.19. Zaťažením motora sa zvýšia prúdy ako je na obrázku 6.18. Rýchlosť motora, po odoznení prechodového deja zaťaženia motora, klesne na hodnotu 15rad/s, avšak kmitá okolo tejto hodnoty. To je spôsobené závislosťou momentu motora na polohe a prúde.



Obrázok 6.19: Uhlová rýchlosť modelu SRM

7. ZÁVER

Práca obsahuje matematický popis asynchrónneho motora, synchronného motora s permanentnými magnetmi, jednosmerného motora s cudzím budením, jednosmerného motora s permanentnými magnetmi, jednosmerného motora so sériovým budením a spínaného reluktančného motora.

Na základe matematických popisov motorov boli vytvorené modely v prostredí Matlab Simulink. Modely boli vytvorené ako so spojitými tak aj s diskretnými stavovými veličinami v rôznych variantoch. Taktiež bola prezentované metóda Space Vector Modulation a na základe popísaného algoritmu [18] bol vytvorený model. Každý model bol doplnený o užívateľské rozhranie vytvorené v programe Matlab cez *Graphical User Interface Layout Editor*.

Na procesore rady Freescale 56F800E Hybrid Controller boli vytvorené modely jednosmerného motora s permanentnými magnetmi a synchronného motora s permanentnými magnetmi.

Funkčnosť modelov bola overená pomocou simulácie a výsledky sú uvedené v kapitole 6. Modely na procesore boli taktiež overené simuláciou v prostredí Matlab Simulink a súčasne porovnané s modelmi vytvorenými priamo v tomto prostredí. Získané odozvy potvrdzujú teoretické predpoklady a správnu funkčnosť jednotlivých modelov.

Z modelov bola vytvorená samostatná knihovňa, s blokmi modelov motorov, transformáciami súradníc a ďalších blokov, ktoré by boli vhodné pre simuláciu modelov. Knihovňa môže poslúžiť ako dobrý nástroj pre simuláciu a testovanie nových riadiacich algoritmov, či vzdelávacie účely. Model na procesore, vybaveného A/D prevodníkmi na vstupe a výstupe, by mohol nahradiť skutočný, cenovo nákladný motor v reálnom regulačnom zapojení, pre testovanie – doladenie regulačného obvodu, za oveľa nižšie cenové náklady.

8. ZOZNAM LITERATÚRY

- [1] *Documentation for MathWorks Products* [citované 10.5.2010]. Dostupné z <<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>>.
- [2] Freescale Semiconductor, Inc., *DSP56800E 16-Bit Digital Signal Procesor Core Reference Mauual* [citované 12.4.2010]. Dostupné z <<http://www.datasheetarchive.com/datasheet-pdf/018/DSA00311367.html>>.
- [3] Freescale Semiconductor, Inc., *CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers: MC56F8xxx/DSP5685x Family Targeting Manual* [citované 9.5.2010]. Dostupné z <http://cache.freescale.com/files/soft_dev_tools/doc/user_guide/Targeting_56800E.pdf>.
- [4] *Alpha Beta Transformation* [citované 8.5.2010]. Dostupné z <http://en.wikipedia.org/wiki/Alpha_beta_transformation>.
- [5] *Dqo transformation* [citované 8.5.2010]. Dostupné z <http://en.wikipedia.org/wiki/Dqo_transformation>.
- [6] CAHA, Z., ČERNÝ M. *Elektrické pohony*. STNL, Praha 1990. 360s. ISBN 80-03-00418-7.
- [7] VÁCLAVEK, P. *Estimace rychlosti pro bezsnímačové řízení elektrických pohonů s asynchronními motory - zkrácená verze habilitační práce*. Brno: VUTIMUM Brno, 2006. 34 s. ISBN 80-214-3314-0
- [8] ENRIQUE L. CARRILLO ARROYO. *Modeling and simulation of permanent magnet synchronous motor drive systém* [citované 18.5.2010]. Dostupné z <<http://grad.uprm.edu/tesis/carrilloarroyo.pdf>>.
- [9] PETRUŠKA, Ľ. *Model synchronného motoru*. (Bakalárska práca) Brno: Vysoké učení technické v Brne, Fakulta elektrotechniky a komunikačních technologií, 2008. 60s. Veducí práce Ing. Libor Veselý.
- [10] Prof. Dr.ISMAIL H. ALTAS. *Dynamic Model of a Permanent Magnet DC Motor*, Kardeniz Tehnical Univrsity, Faculty of Engineering Electrical and Electronics Engineering [citované 24.4.2010]. Dostupné z <http://www.ihaltas.com/courses/fbe_elk5320/eng/projects/Project_37.pdf>

- [11] *Basic DC Electrical Motor Construction* [citované 8.5.2010]. Dostupné z <<http://www.wisc-online.com/objects/ViewObject.aspx?ID=IAU9508>>.
- [12] WOLM P., CHEN X.Q., CHASE J.G., PETTIGREW W., HANN C.E. *Analysis of a PM DC Motor Model of Application in Feedback Design for Electric Powered Mobility Vehicles* [citované 28.4.2010]. Dostupné z <http://ir.canterbury.ac.nz/bitstream/10092/2448/1/12611907_Motor%20Model%20-%20final%20paper.pdf>
- [13] STOLAŘÍK M., VESELKA F. *Přehledová studie reluktančních motorů*. ELEKTRO: odborný časopis, březen 2006, s 4-9. [citované 8.5.2010]. Dostupné z <<http://www.odbornecasopisy.cz/download/el030604.pdf>>
- [14] MAXIM V., ŽIDEK K. *Jednoduchý lineární model spínaného reluktančního motora (SRM)*. AT&P journal PLUS: recenzované periodikum vědeckých a inženýrských publikací, 1/2006 (vychází 2x ročně), s. 45-47. ISSN 1336-5010.
- [15] Freescale Semiconductor, Inc., *3-Phase Switched Reluctance (SR) Sensorless Motor Control Using a 56F80x, 56F8100 or 56F8300 Device*. Application Note: AN1932 Rev. 2, 2/2005, s. 5-12 [citované 12.5.2010]. Dostupné z <<http://cache.freescale.com/files/product/doc/AN1932.pdf>>.
- [16] *Lineární interpolace* [citované 10.5.2010]. Dostupné z <http://cs.wikipedia.org/wiki/Line%C3%A1rn%C3%AD_interpolace>.
- [17] *Bilineární interpolace* [citované 10.5.2010]. Dostupné z <http://cs.wikipedia.org/wiki/Biline%C3%A1rn%C3%AD_interpolace>.
- [18] Interná dokumentácia firmy Freescale

ZOZNAM SKRATIEK A SYMBOLOV

ASM	asynchronný motor
DCM	jednosmerný motor
FEKT	Fakulta elektrotechniky a komunikačních technologií
GUI	grafické uživatelské rozhranie (Graphical User Interface)
LSB	najmenej významný bit (Least Significant Bit)
MSB	najviac významný bit (Most Significant Bit)
PM	permanentné magnety
PMDCM	jednosmerný motor s permanentnými magnetmi
PMSM	synchronný motor s permanentnými magnetmi
PWM	pulzne šírkovou moduláciou (Pulse-width Modulation)
SEPEX	jednosmerný motor s cudzím budením
SWDCM	jednosmerný motor so sériovým vinutím
SRM	spínaný reluktančný motor
SVM	Space Vector Modulation
UAMT	Ústav automatizace a měřicí techniky
a, b, c	riadiace signály
b_i	bit s indexom i
C	konštanta motora
f	frekvencia
f_i	funkcia premennej i
i_a	prúd kotvy
i_a, i_b, i_c	statorové prúdy vo vinutiach a, b, c
$i_\alpha, i_\beta, i_d, i_q$	prúdy v príslušných súradniciach
i_e	prúd budiaceho vinutia
i_f	prúd fáze
$InpXx$	vstupná premenná modelu
$\mathbf{i}_S, \mathbf{i}_R$	komplexor prúdu statora a rotora v T-modeli

$\mathbf{i}_{R\Gamma}, \mathbf{i}'_{R\Gamma}$	komplexor prúdu rotora v Γ -modeli, Γ^{-1} -modeli
J	moment zotrvačnosti motora
K	spoločný súradnicový systém
K_e	konštanta elektrického potenciálu
K_i	konštanty diferenčných rovníc
K_t	momentová konštanta motora
K_{TL}	koefficient tlmenia
K_{tr}	transformačná konštanta
L_a	indukčnosť vinutia kotvy
L_d, L_q	indukčnosti vinutia v d, q osi
L_e	indukčnosť budiaceho vinutia
L_L, L_L'	rozptylová indukčnosť rotora v Γ -modeli, statora v Γ^{-1} -modeli
L_M, L_M'	magnetizačná indukčnosť v Γ -modeli, Γ^{-1} -modeli
L_m	vzájomná indukčnosť v T-modeli
L_S, L_R	indukčnosť vinutia statora, rotora v T-modeli
$L_{S\sigma}, L_{R\sigma}$	rozptylová indukčnosť statora, rotora v T-modeli
M	počet bitov
N	počet parametrov
$OutY_y$	výstupná premenná modelu
O_{xxx}	nulový vektor napätia
P_P	počet pólových dvojíc motora
pwm_0	počet impulzov počítadla pre nulový vektor
pwm_1	počet impulzov počítadla pre vektor č. 1
pwm_2	počet impulzov počítadla pre vektor č. 2
R	súradnicový systém rotora
R_a	odpor vinutia kotvy
R_{ae}	odpor budiaceho vinutia a kotvy
R_e	odpor budiaceho vinutia
R_f	odpor fáze
R_S, R_R	odpor statora, rotora

$R_{RT} R'_{RT}$	odpor rotora Γ -modelu, Γ^{-1} -modelu
s	sklz motora
S	súradnicový systém statora
$shiftK_i$	počet bitov pre bitový posun konštanty K_i
t	čas
T_0, T_1, T_2	časové intervaly pre SVM
T_e	elektrický moment motora
T_{ef}	elektrický moment fáze
T_L	je mechanický moment záťaže
T_{PWM}	perióda vzorkovania PWM
U	množina, vektor vstupov
u_a	napätie kotvy
$U_a U_b U_c$	fázové napätie
$U_{AB} U_{BC} U_{CA}$	združené napätie
$u_\alpha, u_\beta, u_d, u_q$	napätia v príslušných súradniciach
U_{DC}	jednosmerné napätie
u_e	napätie budiaceho vinutia
u_f	napájacie napätie fáze
u_{Lf}	indukované napätie fáze
U_S	amplitúda napätia
$\mathbf{u}_S, \mathbf{u}_R$	komplexor napätia statora a rotora
U_{XXX}	nenulový vektor napätia
$U_X, U_{X \mp 60}$	vektory napätia sektorov
X	množinu stavov
X_c, X_{d_k}	spojité, diskkrétne stavy
x_i, y_i	obecné parametre
x_{rmax}, y_{rmax}	maximálna (prípustná) hodnota
x_{norm}, y_{denorm}	hodnota po procese normalizácie denormalizácie
Y	množina výstupy

γ_f	uhol medzi fázami
γ, γ'	transformačná konštanta Γ -modelu, Γ^{-1} -modelu
ϑ	uhol medzi statorovým a rotorovým vinutím
ϑ^K	uhol medzi spoločným súradnicovým systémom a statorovým súradnicovým systémom
ϑ_e	elektrická poloha
ϑ_m	poloha rotora
$\omega_m, \omega_e, \omega_s$	uhlovou rýchlosť, rotorová (mechanická), elektrická, synchronná
Φ	budiaci tok
θ	poloha rotora
ψ_a, ψ_b, ψ_c	spražený magnetický tok príslušného vinutia
ψ_d, ψ_q	spražený magnetický tok v súradnici d, q
ψ_f	vzájomný sprážený tok medzi statorom a rotorom vyvolaný PM
Ψ_f	magnetický tok fáze
Ψ_S, Ψ_R	spražený magnetické toky statora, rotora
Ψ_S^*	komplexne združený magnetický tok
$\Psi_{R\Gamma}, \Psi'_{R\Gamma}$	spražený magnetický tok rotora v Γ -modeli, Γ^{-1} -modeli

ZOZNAM PRÍLOH

PRÍLOHA A : CD s elektronickou verziou diplomovej práce, s knižnicou vytvorených modelov a zdrojovým kódom modelov vytvorených na procesore rady Freescale 56F800 Hybrid Controller.

PRÍLOHA B : Uživatelská príručka pre prácu s modelmi knihovne *Elektromechanical Drives*